



TUGAS AKHIR - SS141501

**PENENTUAN PANJANG OPTIMAL
DATA DERET WAKTU BEBAS *OUTLIER*
DENGAN METODE *WINDOW TIME***

**RYA SOFI AULIA
NRP 1313100 064**

**Dosen Pembimbing
R. Mohamad Atok, M.Si, Ph.D**

**PROGRAM STUDI S1
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**



TUGAS AKHIR - SS 141501

**PENENTUAN PANJANG OPTIMAL
DATA DERET WAKTU BEBAS *OUTLIER*
DENGAN METODE *WINDOW TIME***

**RYA SOFI AULIA
NRP 1313100 064**

**Dosen Pembimbing
R. Mohamad Atok, M.Si, Ph.D**

**PROGRAM STUDI S1
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**



FINAL PROJECT - SS 141501

***DETERMINATION OF THE OPTIMUM LENGTH
OF FREE OUTLIER TIME SERIES DATA
USING WINDOW TIME METHODS***

**RYA SOFI AULIA
NRP 1313100 064**

**Supervisor
R. Mohamad Atok, M.Si, Ph.D**

**UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**

LEMBAR PENGESAHAN

PENENTUAN PANJANG OPTIMAL DATA DERET WAKTU BEBAS *OUTLIER* DENGAN MENGUNAKAN METODE *WINDOW TIME*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
Pada

Program Studi S-1 Jurusan Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember

Oleh:

RYA SOFI AULIA
NRP. 1313 100 064

Disetujui oleh Pembimbing Tugas Akhir:

R. Mohamad Atok, M.Si, Ph.D ()
NIP. 19710915 199702 1 001

Mengetahui,
Ketua Jurusan Statistika FMIPA-ITS




Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JANUARI 2017

PENENTUAN PANJANG OPTIMAL DATA DERET WAKTU BEBAS *OUTLIER* DENGAN MENGGUNAKAN METODE *WINDOW TIME*

Nama Mahasiswa : Rya Sofi Aulia
NRP : 1313 100 064
Jurusan : Statistika FMIPA - ITS
Dosen Pembimbing: R. Mohamad Atok, M.Si, Ph.D

Abstrak

Data outlier sering kali mempengaruhi model data secara umum sehingga pengaruh dari data outlier tersebut harus dikurangi atau dihilangkan. Namun, di sisi lain outlier merupakan data yang sangat informatif apabila penyebab adanya outlier tersebut diketahui sehingga beberapa penelitian merekomendasikan untuk tidak menghilangkan outlier namun mengganti model awal dengan model baru yang disisipkan dengan model outlier. Kemunculan outlier dapat menyebabkan bias yang cukup serius dalam estimasi parameter. Atas dasar penelitian-penelitian yang dilakukan sebelumnya maka pada penelitian ini dilakukan metode baru untuk mendeteksi outlier. Tujuan dari metode ini adalah untuk mendapatkan panjang data optimum yang bisa digunakan untuk mendeteksi data outlier. Penelitian ini terfokus pada pendeteksian outlier pada data deret waktu dengan jumlah data yang banyak. Dari hasil simulasi data dan implementasi yang dilakukan pada data riil didapatkan hasil bahwa window time 500 dan 1000 memberikan nilai akurasi deteksi outlier lebih baik dibandingkan dengan window time 100. Selain itu, metode deteksi menggunakan window time memberikan hasil yang lebih baik dibandingkan metode deteksi outlier biasa.

Kata Kunci : Data Bebas Outlier, Outlier, Window Time

(halaman ini sengaja dikosongkan)

DETERMINATION OF THE OPTIMUM LENGTH OF FREE OUTLIER TIME SERIES DATA USING WINDOW TIME METHODS

Name : Rya Sofi Aulia
NRP : 1313 100 064
Department : Statistics FMIPA - ITS
Supervisor : R. Mohamad Atok, M.Si, Ph.D

Abstract

Data outliers often affects the common data model so that the effect of the outlier data is to be reduced or eliminated. However, on the other hand outlier is the data that is very informative if the cause of the outlier is known that several studies recommend not eliminate outliers but replace the initial model with a new model that is inserted with the model outliers. The emergence of outliers can cause quite serious bias in the estimation of parameters. Based on the studies conducted previously, this research carried out new methods for detecting outliers. The purpose of this method is to obtain the optimum length of the data that can be used to detect the data outliers. This research is focused on outlier detection in time series data with large amounts of data. From the simulation results and the implementation of data on real data showed that window time 500 and 1000 deliver the accuracy of outlier detection is better than the window time 100. In addition, the detection method using the window time gives better results than usual outlier detection method.

Keywords: *Free Data Outliers, Outliers, Window Time*

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur yang kehadirat Allah SWT, Tuhan Yang Maha Esa. Berkat rahmat dan ridho-Nya penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“Penentuan Panjang Optimal Data Deret Waktu Bebas *Outlier* Menggunakan Metode *Window Time*”** dengan lancar.

Keberhasilan penyusunan Tugas Akhir ini tidak lepas dari banyaknya bantuan dan dukungan yang diberikan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Suhartono selaku Ketua Jurusan Statistika dan Bapak Dr. Sutikno, M.Si selaku Koordinator Program Studi S1 yang telah memberikan fasilitas untuk kelancaran penyelesaian Tugas Akhir.
2. Bapak R. Mohamad Atok, M.Si, Ph.D selaku dosen pembimbing yang telah dengan sabar memberikan bimbingan, saran, dan dukungan selama penyusunan Tugas Akhir.
3. Ibu Dr. Kartika Fithriasari, M.Si dan Bapak Dr. Ir. Setiawan, M.S selaku dosen penguji yang telah memberikan banyak bantuan dan saran untuk kesempurnaan Tugas Akhir ini.
4. Bapak Dr. Agus Suharsono, M.S selaku dosen wali yang telah memberikan nasehat dan semangat.
5. Seluruh dosen Statistika ITS yang telah memberikan ilmu dan pengetahuan yang tak ternilai harganya, serta segenap karyawan Jurusan Statistika ITS.
6. Ida Zulaicha dan Ansori yaitu Ibu dan Ayah penulis yang selalu memberikan dukungan, kasih sayang dan doa yang tidak pernah putus, serta senantiasa menjadi penyemangat bagi penulis dalam menyelesaikan Tugas Akhir ini.
7. Nay, Dwi, Ratih, Hana, Ochid, Enis, Yoshi, Adheala yang telah saling membantu dan memberikan semangat.

8. Hendra yang juga senantiasa memberikan semangat, membantu dan mendoakan penulis.
9. Dek Rima dan Dek Fara selaku adik sepupu penulis yang selalu memberikan dukungan, bantuan dan semangat kepada penulis.
10. Mas Ahmad dan Bani yang sudah bersedia meluangkan waktu untuk berdiskusi mengenai topik Tugas Akhir ini.
11. Irma yaitu teman sekamar penulis yang selalu saling mengingatkan penulis dan membantu selama ini.
12. Mbak Fefy, Mbak Jupita, Mbak Ulfa, Mas Afrian, Mas Rohim, Mas Amma dan Mas Ivan yaitu pengurus KOPMA dr. Angka ITS 2015 yang selalu memberikan motivasi untuk menyelesaikan Tugas Akhir ini.
13. Beasiswa Bidikmisi yang telah mendukung perkuliahan penulis selama ini.
14. Teman-teman Pejuang 115 atas semangat yang selalu diberikan kepada penulis.
15. Teman-teman Sigma 24 yang selalu memberikan kehangatan dan kenyamanan kepada penulis selama ini.
16. Semua pihak yang telah memberikan bantuan hingga penyusunan laporan Tugas Akhir ini dapat terselesaikan.

Penulis berharap hasil Tugas Akhir ini dapat bermanfaat bagi kita semua. Semoga kebaikan dan bantuan yang telah diberikan kepada penulis dibalas dengan kebaikan yang lebih besar lagi oleh Tuhan Yang Maha Esa. Aamiin.

Surabaya, Januari 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
COVER PAGE	iii
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR LAMPIRAN	xxiii
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	5
1.5 Batasan Penelitian	6
BAB II TINJAUAN PUSTAKA	
2.1 Analisis <i>Time Series</i>	7
2.2 Model ARIMA	10
2.2.1 Model <i>Autoregressive</i> (AR)	10
2.2.2 Model <i>Moving Average</i> (MA).....	11
2.3 Evaluasi Model.....	17
2.4 Deteksi Outlier	17
2.4.1 <i>Additional Outlier</i> (AO).....	18
2.4.2 <i>Innovational Outlier</i> (IO).....	18
2.4.3 <i>Temporary Change</i> (TC).....	19
2.4.4 <i>Level Shift</i> (LS)	19
2.4.5 Metode Pendeteksian Outlier Chen dan Liu (1993).....	20
2.5 Metode <i>Window Time</i>	24

2.5.1	<i>Window Time Full Memory dan No Memory</i>	24
2.5.2	<i>Window Time Fixed Size dan Adaptable Size</i>	25
2.5.3	<i>Metode Batch Selection</i>	27
2.6	Uji ANOVA	28
2.6.1	Uji Asumsi Kenormalan	28
2.6.2	Uji Asumsi Homogenitas	29
2.6.3	Uji ANOVA	29

BAB III METODOLOGI PENELITIAN

3.1	Sumber Data	31
3.2	Langkah Analisis	31
3.3	Diagram Alir	34

BAB IV ANALISIS DAN PEMBAHASAN

4.1	Data Simulasi	37
4.2	Penyisipan Outlier	38
4.2.1	Pengaruh <i>Additional Outlier</i> (AO)	39
4.2.2	Pengaruh <i>Innovational Outlier</i> (IO)	39
4.2.3	Pengaruh <i>Temporary Change</i> (TC)	40
4.2.4	Pengaruh <i>Level Shift</i> (LS)	41
4.3	Prosedur Deteksi <i>Outlier</i> Dengan Metode <i>Window Time</i>	42
4.3.1	Prosedur Deteksi <i>Outlier</i> Dengan <i>Window Time</i> Awal 100	44
4.3.2	Prosedur Deteksi <i>Outlier</i> Dengan <i>Window Time</i> Awal 500	49
4.3.3	Prosedur Deteksi <i>Outlier</i> Dengan <i>Window Time</i> Awal 1000	54
4.3.4	Pembahasan Outlier Jenis <i>Level Shift</i>	59
4.3.5	Pengujian Multi Way ANOVA Untuk Mengetahui Pengaruh Lebar <i>Window Time</i> Awal	60
4.4	Membandingkan Akurasi Hasil Prediksi	65
4.5	Studi Kasus (<i>Tree Rings</i>)	66
4.5.1	Identifikasi Model	67
4.5.2	Estimasi dan Signifikansi Parameter	70
4.5.3	Diagnostic Checking	70
4.5.4	Hasil Prediksi	71

BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan.....	73
5.2 Saran	74
DAFTAR PUSTAKA	75
LAMPIRAN	79
BIODATA PENULIS.....	101

(halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Plot Time Series Data Stasioner dalam Rata-rata dan Varians	8
Gambar 2.2 Plot ACF Data Stasioner.....	9
Gambar 2.3 Plot ACF Data Tidak Stasioner	9
Gambar 2.4 <i>Full Memory dan No Memory Time Window</i>	25
Gambar 2.5 <i>Fixed Size Time Window</i>	26
Gambar 2.6 <i>Adaptable Size Time Window</i>	27
Gambar 2.7 <i>Batch Selection</i>	28
Gambar 3.1 Diagram Alir Langkah Simulasi Data	34
Gambar 3.2 Diagram Alir Langkah Penyisipan Outlier dan Proses Deteksi Outlier.....	35
Gambar 3.3 Diagram Alir Langkah Deteksi Outlier Chen & Liu (1993).....	36
Gambar 4.1 <i>Time Series Plot Model Simulasi Perulangan Pertama</i>	38
Gambar 4.2 <i>Time Series Plot Model $Z_t = -0,8Z_{t-1} + a_t$ Perulangan Pertama Setelah Penambahan AO</i> ...	39
Gambar 4.3 <i>Time Series Plot Model $Z_t = -0,8Z_{t-1} + a_t$ Perulangan Pertama Setelah Penambahan IO</i>	40
Gambar 4.4 <i>Time Series Plot Model $Z_t = -0,8Z_{t-1} + a_t$ Perulangan Pertama Setelah Penambahan TC</i>	41
Gambar 4.5 <i>Time Series Plot Model $Z_t = -0,8Z_{t-1} + a_t$ Perulangan Pertama Setelah Penambahan LS</i>	42
Gambar 4.6 Ilustrasi Pembagian <i>Window Time</i> dengan Lebar Window Awal.....	44
Gambar 4.7 <i>Time Series Plot Data Tree Rings</i>	67
Gambar 4.8 <i>Box-Cox Plot Data Tree Rings</i>	68

Gambar 4.9 *ACF Plot Data Tree Rings*68

Gambar 4.10 *PACF Plot Data Tree Rings*.....69

Gambar 4.11 *Residual Plot Data Tree Rings*.....70

DAFTAR TABEL

	Halaman
Tabel 2.1	Nilai Konstanta λ dan Fungsi Transformasinya12
Tabel 2.2	Pola Plot ACF dan PACF untuk Model Non Musiman13
Tabel 2.3	Tabel <i>Multi Way ANOVA</i>30
Tabel 4.1	Empat Model yang Digunakan Dalam Simulasi.....37
Tabel 4.2	Prosentase Kesalahan Deteksi <i>Outlier</i> AO dengan Panjang <i>Initial Window</i> 10047
Tabel 4.3	Prosentase Kesalahan Deteksi <i>Outlier</i> IO dengan Panjang <i>Initial Window</i> 10048
Tabel 4.4	Prosentase Kesalahan Deteksi <i>Outlier</i> TC dengan Panjang <i>Initial Window</i> 10049
Tabel 4.5	Prosentase Kesalahan Deteksi <i>Outlier</i> AO dengan Panjang <i>Initial Window</i> 50052
Tabel 4.6	Prosentase Kesalahan Deteksi <i>Outlier</i> IO dengan Panjang <i>Initial Window</i> 50053
Tabel 4.7	Prosentase Kesalahan Deteksi <i>Outlier</i> TC dengan Panjang <i>Initial Window</i> 50054
Tabel 4.8	Prosentase Kesalahan Deteksi <i>Outlier</i> AO dengan Panjang <i>Initial Window</i> 100057
Tabel 4.9	Prosentase Kesalahan Deteksi <i>Outlier</i> IO dengan Panjang <i>Initial Window</i> 100058
Tabel 4.10	Prosentase Kesalahan Deteksi <i>Outlier</i> TC dengan Panjang <i>Initial Window</i> 100059
Tabel 4.11	Rata-Rata Kesalahan Deteksi <i>Outlier</i> Berdasarkan Lebar <i>Window Time</i> Awal61
Tabel 4.12	<i>Levene's Test</i> Untuk Menguji Homogenitas61
Tabel 4.13	Hasil <i>Multi Way ANOVA</i>62
Tabel 4.14	Hasil <i>Uji Tukey Post Hoc</i>64

Tabel 4.15 Hasil Perbandingan Nilai RMSE 65

Tabel 4.16 *Augmented Dickey-Fuller Test* Untuk Menguji
Stasioneritas Terhadap *Mean* 69

Tabel 4.17 Signifikansi Parameter 69

Tabel 4.18 Pengecekan Residual *White Noise* 70

Tabel 4.19 Perbandingan RMSE Ketiga Cara Pada Data
Tree Rings 71

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Statistika Deskriptif Prosentase Kesalahan Deteksi <i>Outlier</i>	77
Lampiran 2 Hasil Pengujian ANOVA Prosentase Kesalahan Deteksi <i>Outlier</i>	89
Lampiran 3 Hasil Pengujian <i>Tukey</i> Prosentase Kesalahan Deteksi <i>Outlier</i>	90
Lampiran 4 Data Simulasi Model ARIMA (1,0,0) dengan Parameter $\phi = 0,8$	91
Lampiran 5 Data Simulasi Model ARIMA (1,0,0) dengan Parameter $\phi = -0,8$	92
Lampiran 6 Data Simulasi Model ARIMA (1,0,0) dengan Parameter $\phi = 0,5$	93
Lampiran 7 Data Simulasi Model ARIMA (1,0,0) dengan Parameter $\phi = -0,5$	94
Lampiran 8 Prosentase Kesalahan Deteksi <i>Outlier</i> Kombinasi Parameter, Jenis <i>Outlier</i> , Lebar <i>Window Time</i> Awal dan Lokasi <i>Outlier</i>	95
Lampiran 9 Parameter Data Simulasi Model ARIMA (1,0,0)	96
Lampiran 10 Data <i>Tree Rings</i>	97
Lampiran 11 Surat Pernyataan Data Tugas Akhir	98

(halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Model *time series* secara umum digunakan untuk mempelajari kehomogenan pola *memory* pada data *time series*. Keberadaan data *outliers* maupun perubahan struktural data menurunkan efisiensi dalam estimasi model *autoregressive* (AR). *Outlier* dan perubahan struktural data merupakan suatu hal yang umum ditemui dalam analisis data *time series* sehingga dapat menghasilkan kesimpulan yang salah. Data *outlier* merupakan data observasi yang memiliki karakteristik yang berbeda dengan data lainnya. *Outlier* dibedakan menjadi 4 jenis yaitu *Additional Outlier* (AO), *Innovation Outlier* (IO), *Temporary Change* (TC) dan *Level Shift* (LS). *Outlier* dengan jenis AO dan IO merupakan jenis *outlier* yang khusus sedangkan LC dan TC merupakan *outlier* yang mengalami perubahan titik atau struktural.

Data *outlier* sering kali mempengaruhi model data secara umum sehingga pengaruh dari data *outlier* tersebut harus dikurangi atau dihilangkan. Di sisi lain, *outlier* merupakan data yang sangat informatif apabila penyebab adanya *outlier* tersebut diketahui sehingga beberapa penelitian merekomendasikan untuk tidak menghilangkan *outlier* namun mengganti model awal dengan model baru yang disisipkan dengan model *outlier*.

Untuk mengidentifikasi model parameter yang paling baik, maka data-data *outlier* harus dideteksi dengan cara menghilangkan pengaruh *outlier* maupun menghilangkan data *outlier* tersebut. Berbagai macam metode pendeteksian *outlier* telah dicobakan oleh beberapa peneliti.

Tsay (1986) melakukan penelitian mengenai spesifikasi model *time series* ketika ditemukan *outlier* pada data deret waktu. Data *outlier* merupakan suatu kejadian yang wajar terjadi dan sering kali muncul dalam analisis data, termasuk data *time series*. Pengaruh dari adanya data *outlier* bisa menyebabkan bias atau salah prediksi pada model data *time series* tersebut. Oleh karena

itu, sebelum dilakukan analisis *time series* diperlukan deteksi *outlier* terlebih dahulu untuk mengetahui keberadaan *outlier* dan tindakan apa yang akan dilakukan pada data *outlier* tersebut. Pengeliminasian *outlier* adalah salah satu cara yang digunakan untuk menghapus data *outlier* sehingga spesifikasi model yang terbentuk dapat sebaik mungkin. Metode yang digunakan merupakan prosedur iterasi untuk mengidentifikasi *outlier*, kemudian menghilangkan pengaruh *outlier* tersebut dan akhirnya menentukan model sementara yang terbentuk setelah pengaruh *outlier* dikeluarkan.

Kemudian Tsay (1988) kembali melakukan penelitian tentang *outliers*, *level shift* dan perubahan varians dalam data deret waktu. Ketiga jenis kejadian ini mempengaruhi stabilitas model *time series*. Namun terkadang keberadaannya sering diabaikan dan pengaruhnya diremehkan dampaknya. Oleh karena itu diperlukan metode yang berguna untuk mendeteksi dan mengatasi keberadaan kejadian yang berbeda dengan data pada umumnya. Metode yang digunakan cukup sederhana dengan menggunakan *least square* dan rasio varians residual. Berbagai masalah yang muncul dalam metode deteksi *outlier*, *level shift* dan perubahan varians juga tetap harus dipertimbangkan. Dalam penelitian ini dilakukan percobaan pada tiga data real untuk menguji keefektifan prosedur deteksi *outlier* tersebut.

Parameter dari model *time series* dan pengaruh *outlier* dapat pula diestimasi secara bersama (Chen & Liu, 1993). *Outliers* merupakan data yang kemunculannya tidak bisa diprediksi karena terdapat berbagai macam faktor yang dapat menjadi penyebab munculnya *outlier* tersebut. *Outlier* dapat memberikan pengaruh yang cukup signifikan pada hasil identifikasi, estimasi parameter dan hasil peramalan. Metode yang digunakan adalah deteksi *outlier* secara iteratif untuk mendapatkan estimasi parameter dari model *time series* dan pengaruh *outlier* secara bersama. Pada percobaannya menggunakan 4 jenis *outlier* yang terdapat dalam data deret waktu yaitu AO, IO, TC dan LC. Perbedaan yang mendasar antara metode yang digunakan oleh Chen & Liu (1993)

dengan literatur sebelumnya adalah (a) jenis *outlier* menimbulkan dampak yang tidak terlalu signifikan terhadap spesifikasi model, (b) pengaruh *outlier* yang diestimasi menggunakan regresi berganda, dan (c) parameter model dan pengaruh *outlier* diestimasi secara bersama. Data yang digunakan untuk memeriksa statistik uji pada data dengan panjang yang berbeda-beda dilakukan dengan data simulasi. Cara yang digunakan oleh Chen & Liu (1993) bekerja dengan baik untuk mendeteksi *outlier* dan mendapatkan estimasi parameter yang tidak bias. Cara ini kemudian diaplikasikan pada data real dan menghasilkan performansi yang efektif dalam menghindari deteksi data *outlier* padahal data tersebut bukan *outlier*. Estimasi parameter model yang didapatkan dari cara tersebut mirip dengan metode yang menggunakan maksimum *likelihood* dengan model intervensi untuk memasukkan *outlier*.

Atok, et al. (2015) melakukan penelitian tentang pendeteksian perubahan sementara pada model data ARMA(1,1). Penelitian tersebut menggunakan data simulasi dengan model ARMA (1,1) dengan 4 macam kombinasi parameter yang berbeda. Residual dihasilkan dengan menggunakan metode *Conditional Least Square* (CLS) dan *Median Absolute Deviation* (MAD). Pengaruh *outlier* diatasi dengan menggunakan dua cara (a) mengganti data *outlier* dengan nilai data lain yang bukan *outlier* dan (b) membuang data *outlier*. Pada semua kasus, cara tersebut memberikan hasil terbaik dengan membuang data *outlier* dengan estimasi residual menggunakan metode *Median Absolute Deviation* (MAD). Metode ini diimplementasikan pada data polusi udara di Surabaya dan memberikan hasil yang sama yaitu metode terbaik dilakukan dengan cara mengeliminasi data *outlier* dan mengestimasi residual dengan menggunakan metode *Median Absolute Deviation* (MAD).

Pada metode deteksi *outlier* yang dilakukan oleh peneliti-peneliti sebelumnya, *outlier* yang terkandung di dalam suatu data dapat dideteksi dengan menggunakan hasil spesifikasi model yang masih mengandung *outlier* sehingga bisa terjadi kesalahan hasil

prediksi keberadaan *outlier* serta hasil *forecasting*-nya. Namun, pada penelitian yang akan dilakukan ini spesifikasi model dibangun dari data yang bebas *outlier* sehingga diharapkan dapat meningkatkan keakuratan hasil deteksi *outlier*.

Selain melakukan deteksi *outlier* dengan menggunakan keseluruhan data, dapat dilakukan dengan cara pemodelan *window time* yaitu memodelkan dengan semua data *in sampel* kemudian model yang diperoleh akan digunakan pada masing-masing *window time* yang telah dibentuk (Hadi, 2016). Misalnya digunakan data *in sampel* sebanyak 11 tahun, kemudian spesifikasi model yang diperoleh dari keseluruhan data tersebut digunakan pada data dengan lebar *window time* 4 tahun, 5 tahun, ..., 10 tahun. Berpedoman pada cara tersebut, deteksi *outlier* dengan pembagian *window time* dapat dilakukan dengan cara yang sama. Misalnya, data *in sampel* yang digunakan sebanyak 4800 data, kemudian model yang diperoleh dari data tersebut digunakan untuk memprediksi keberadaan *outlier* pada 100 data terakhir. Apabila terdapat *outlier*, maka *outlier* tersebut dihilangkan, namun apabila tidak ada *outlier* maka 100 data terakhir yang bebas *outlier* tersebut dimodelkan untuk memprediksi keberadaan 200 data terakhir, dan seterusnya.

Kemunculan *outlier* dapat menyebabkan bias yang cukup serius dalam estimasi parameter model AR, MA dan ARMA. Atas dasar penelitian-penelitian yang dilakukan sebelumnya maka pada penelitian ini dilakukan prosedur baru untuk mendeteksi *outlier* yang ada pada data deret waktu sehingga nantinya akan diperoleh panjang data optimum yang bisa digunakan untuk mendeteksi data *outlier* pada data deret waktu dengan jumlah data yang banyak. Selain itu juga diperoleh lokasi data optimum yang paling baik untuk mendeteksi maupun melakukan peramalan data yang akan datang dengan syarat data tersebut mengandung *outlier* di dalamnya. Dasar teori yang digunakan sebagai landasan penelitian ini akan dibahas pada Bab II. Dan Bab III akan menjelaskan tentang metodologi penelitian dan tahapan yang digunakan dalam prosedur deteksi *outlier*.

Setelah proses simulasi data dilakukan sampai prosedur panjang dan lokasi optimum data bebas *outlier* berhasil didapatkan, kemudian prosedur tersebut diimplementasikan pada data *tree rings* yang didapatkan dari *Time Series Data Library*. Data ini merupakan informasi yang menyediakan lingkaran pohon yang terdapat di hutan yang ada di Chili. Data ini akan diterapkan untuk membandingkan antara deteksi *outlier* yang digunakan sebelumnya dan deteksi *outlier* yang diusulkan dalam penelitian ini. Selain itu data ini juga dapat dicobakan untuk prosedur penentuan panjang optimum data deret waktu bebas *outlier*.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang yang sudah disampaikan pada sub bab sebelumnya, maka permasalahan utama yang akan diteliti adalah prosedur untuk mendapatkan potongan data optimum dari keseluruhan data deret waktu yang bebas dari berbagai jenis *outlier*.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang sudah dijelaskan pada sub bab sebelumnya, maka tujuan yang akan dicapai adalah mendapatkan panjang optimal data yang dibutuhkan untuk memprediksi suatu data deret waktu bebas *outlier* dengan model ARIMA (1,0,0).

1.4 Manfaat Penelitian

Manfaat yang ingin dicapai pada penelitian ini adalah sebagai berikut:

Hasil penelitian ini diharapkan dapat memberikan acuan kepada peneliti yang menggunakan analisis time series selanjutnya mengenai panjang data yang optimum untuk mendapatkan data deret waktu yang bebas *outlier*.

1.5 Batasan Penelitian

Batasan masalah dalam penelitian ini adalah model yang diteliti untuk mengetahui panjang data dan lokasi data optimal

adalah model ARIMA (1,0,0) dengan parameter $\phi=0,8; -0,8; 0,5$ dan $-0,5$. Nilai-nilai parameter ini digunakan karena ingin membandingkan hasil yang diberikan oleh parameter yang bernilai positif dan negatif serta nilai parameter yang mendekati maksimal yaitu 0.8 serta nilai parameter yang intermediate yaitu 0.5. Jumlah *outlier* yang diujikan dalam masing-masing model simulasi adalah *outlier* tunggal (*single outlier*). *Window time* awal yang diujicobakan adalah 100, 500 dan 1000. Alasan pemilihan lebar *window time* tersebut karena ingin membandingkan antara lebar *window time* yang pendek, sedang dan panjang. *Critical value* yang direkomendasikan oleh Chang dan Tiao (1983) untuk data yang panjang adalah 3,0; 3,5 dan 4,0 namun dalam penelitian ini digunakan *critical value* 4,0 supaya memberikan hasil deteksi *outlier* yang lebih akurat.

BAB II TINJAUAN PUSTAKA

2.1 Analisis Time Series

Analisis *time series* diperkenalkan pada tahun 1970 oleh George E. P. Box dan Gwilym M. Jenkins melalui bukunya *Time Series Analysis: Forecasting and Control*. Sejak saat itu, *time series* mulai banyak dikembangkan. Dasar pemikiran *time series* adalah pengamatan sekarang (Z_t) tergantung pada satu atau beberapa pengamatan sebelumnya (Z_{t-k}). Dengan kata lain, model *time series* dibuat karena secara statistik ada korelasi antar deret pengamatan. Untuk melihat adanya korelasi antar pengamatan, dapat dilakukan uji korelasi antar pengamatan yang sering dikenal dengan *Autocorrelation Function* (ACF). Tujuan analisis *time series* antara lain memahami dan menjelaskan mekanisme tertentu, meramalkan suatu nilai di masa depan, dan mengoptimalkan sistem kendali. Analisis *time series* dapat diterapkan di bidang ekonomi, bisnis, industri, teknik dan ilmu-ilmu sosial (Makridakis, 1992).

Berbagai metode telah dikembangkan dalam mengolah data *time series* untuk memperoleh suatu model yang memberikan hasil ramalan yang lebih akurat. Metode yang digunakan antara lain adalah metode ARIMA Box-Jenkins (Box G. J., 1994) yang digunakan untuk mengolah *time series* yang univariat dan metode analisis fungsi transfer digunakan untuk mengolah data *time series* multivariat. Di dalam pembentukan model fungsi transfer digunakan metode ARIMA Box-Jenkins untuk menggabungkan deret-deret input (Z_t) dan input-input lain yang digabungkan dalam satu kelompok yang disebut *noise* (n_t). Untuk dapat diolah dengan menggunakan metode ARIMA Box-Jenkins, suatu data *time series* harus memenuhi syarat stasioneritas.

Misal Z_1, Z_2, \dots, Z_t merupakan proses stokastik untuk runtun waktu diskrit. Proses di atas disebut stasioner jika mean dan

variansinya konstan untuk setiap titik t dan kovarian yang konstan untuk setiap selang waktu ke- k

$$E(Z_t) = \mu \quad \text{konstan untuk semua } t$$

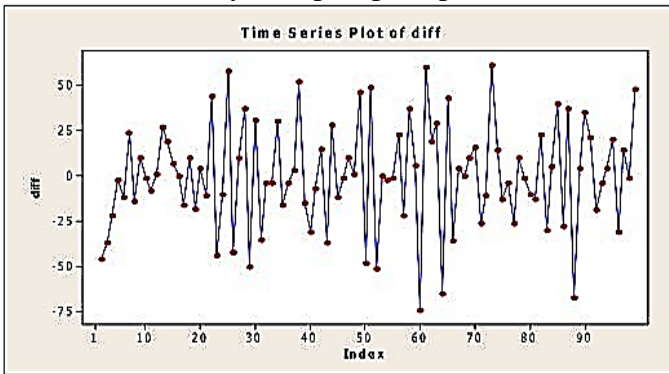
$$\text{Var}(Z_t) = \sigma^2 \quad \text{konstan untuk semua } t$$

$$\text{Cov}(Z_t, Z_{t-k}) = \gamma_k \quad \text{konstan untuk semua } t \text{ dan semua } k \neq 0$$

γ_k adalah autokovariansi pada lag- k .

(Soejoeti, 1987).

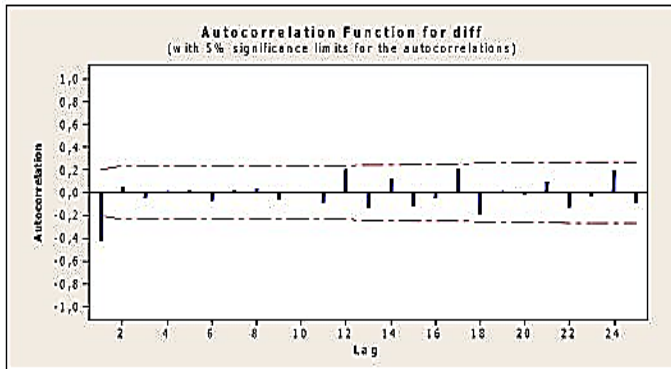
Stasioneritas berarti bahwa tidak terjadi pertumbuhan dan penurunan data. Suatu data dapat dikatakan stasioner apabila pola data tersebut berada pada kesetimbangan di sekitar nilai rata-rata yang konstan dan variansi di sekitar rata-rata tersebut konstan selama waktu tertentu (Makridakis, 1992). *Time series* dikatakan stasioner apabila tidak ada unsur *trend* dalam data dan tidak ada unsur musiman atau rata-rata dan variansinya tetap, seperti pada Gambar 2.1.



Gambar 2.1 Plot Time Series Data Stasioner dalam Rata-rata dan Varians (Hanke & Wichern, 2005)

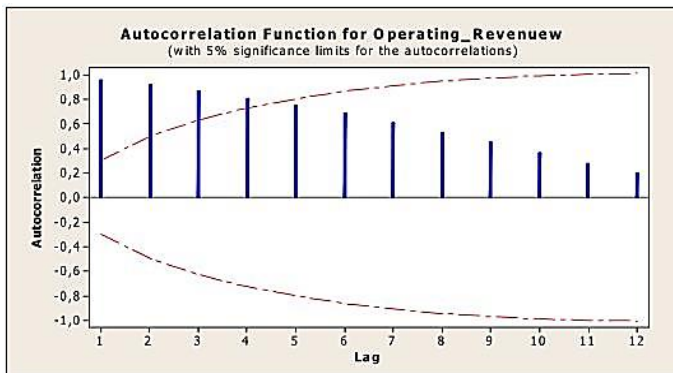
Selain dari plot *time series*, stasioner dapat dilihat dari plot *autocorrelation function* (ACF) data tersebut. Apabila plot data *autocorrelation function* (ACF) turun mendekati nol secara cepat, pada umumnya setelah lag kedua atau ketiga maka dapat dikatakan

stasioner (Hanke & Wichern, 2005) Gambar 2.2 menunjukkan plot ACF dari data stasioner.



Gambar 2.2 Plot ACF Data Stasioner (Hanke & Wichern, 2005)

Data non-stasioner apabila terdapat unsur *trend* dalam data, yaitu mengalami kenaikan dan penurunan seiring bertambahnya periode waktu. Pada data non-stasioner yang memiliki *trend* akan memiliki nilai *Autocorrelation Function* (ACF) yang signifikan pada lag-lag awal kemudian turun secara lambat, seperti Gambar 2.3.



Gambar 2.3 Plot ACF Data Tidak Stasioner (Hanke & Wichern, 2005)

2.2 Model ARIMA

Model *Autoregressive Integrated Moving Average* (ARIMA) merupakan model ARMA nonstasioner yang telah di-*differencing* sehingga menjadi model stasioner. Model ARIMA yang stasioner dan *invertible* dapat dituliskan:

$$\phi(B)Z_t = \theta(B)a_t \quad (1)$$

dimana

$$\phi(B) = (1 - \phi B - \phi^2 B^2 - \dots - \phi^p B^p), \quad \theta(B) = (1 - \theta B - \theta^2 B^2 - \dots - \theta^q B^q)$$

B adalah operator backshift dan a_t adalah residual *white noise*.

Persamaan 1 dapat ditulis sebagai:

$$Z_t = \frac{\theta(B)}{\phi(B)} a_t \quad (2)$$

Ada beberapa model ARIMA yang dapat digunakan pada data *time series*, yaitu:

2.2.1 Model Autoregressive (AR)

Model *Autoregressive* (AR) dengan order p dinotasikan dengan AR(p). Bentuk umum model AR(p) adalah:

$$Z_t = \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + a_t \quad (3)$$

dengan

Z_t : nilai variabel pada waktu ke- t

ϕ_i : koefisien *autoregressive*, $i=1,2,3,\dots,p$

a_t : nilai residual pada waktu ke- t

p : order AR

Persamaan di atas dapat ditulis menggunakan operator B (*backshift*):

$$Z_t = \phi_1 B Z_t + \dots + \phi_p B^p Z_t + a_t \quad (4)$$

$$\phi_1(B)Z_t = a_t \quad (5)$$

2.2.2 Model Moving Average (MA)

Moving Average (MA) merupakan nilai *time series* pada waktu t yang dipengaruhi oleh unsur kesalahan pada saat ini dan unsur kesalahan terbobot pada masa lalu (Makridakis, 1992)

Model *Moving Average* (MA) order q , dinotasikan menjadi MA (q). Secara umum, model MA (q) adalah:

$$Z_t = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (6)$$

dengan

Z_t : nilai variabel pada waktu ke- t

θ_i : parameter model *moving average* (MA)

a_t : nilai galat pada waktu ke- t

q : order MA

Persamaan di atas dapat ditulis menggunakan operator B:

$$Z_t = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) a_t \quad (7)$$

$$Z_t = \theta(B) a_t \quad (8)$$

dan $\theta(B) = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)$ merupakan operator MA.

1. Identifikasi Model

Pada tahap ini akan dilakukan identifikasi model dalam rangka mengetahui order dari orde ARIMA (p, d, q). Namun, sebelum melangkah lebih jauh, perlu dipastikan bahwa data yang digunakan telah memenuhi asumsi stasioneritas dalam rata-rata maupun varian. Data yang masih belum memenuhi asumsi stasioneritas dalam rata-rata perlu dilakukan *differencing* menggunakan rumus pada persamaan (9).

$$W_t = (1 - B)^d Y_t \quad (9)$$

Kemudian jika data juga belum memenuhi asumsi stasioneritas dalam varian maka data perlu ditransformasi menggunakan transformasi Box-Cox. Persamaan yang digunakan dalam transformasi Box-Cox adalah:

$$T(Y_t) = \begin{cases} \frac{Y_t^\lambda - 1}{\lambda}, \lambda \neq 0 \\ \log(Y_t), \lambda = 0, \end{cases} \quad (10)$$

dimana λ merupakan parameter pada transformasi Box-Cox (Box & Cox, 1964). Nilai konstanta λ beserta dengan fungsi transformasi yang sering digunakan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Nilai Konstanta λ dan Fungsi Transformasinya

Konstanta λ	Fungsi Transformasi
-1,0	$\frac{1}{Y_t}$
-0,5	$\frac{1}{\sqrt{Y_t}}$
0,0	$\ln Y_t$
0,5	$\sqrt{Y_t}$
1,0	Y_t

Pada proses identifikasi ini diperlukan plot dari *Autocorrelation Function* (ACF) dan *Partial Autocorrelation Function* (PACF) dari data yang telah stasioner untuk mengetahui dan menentukan orde ARIMA yang tepat. ACF merupakan fungsi korelasi antara Y_t dan Y_{t+k} dengan rumus seperti pada persamaan (2.13).

$$\begin{aligned} \hat{\rho}_k &= \widehat{Corr}(Y_t, Y_{t-k}) \\ &= \frac{\widehat{Cov}(Y_t, Y_{t-k})}{\sqrt{\widehat{Var}(Y_t)}\sqrt{\widehat{Var}(Y_{t-k})}} \\ &= \frac{\sum_{t=k}^T (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^T (Y_t - \bar{Y})^2}. \end{aligned} \quad (11)$$

PACF merupakan fungsi korelasi antara Y_t dan Y_{t-k} dengan mengeluarkan dependensi linier $Y_{t-1}, Y_{t-2}, \dots, Y_{t-k-1}$ atau $Corr(Y_t, Y_{t-k} | Y_{t-1}, Y_{t-2}, \dots, Y_{t-k-1})$ (Wei W. W., 2006). Persamaan untuk mendapatkan nilai PACF adalah sebagai berikut :

$$\begin{aligned}\hat{\phi}_{k+1,k+1} &= \text{Corr}(Y_t, Y_{t-k} \mid Y_{t-1}, Y_{t-2}, \dots, Y_{t-k-1}) \\ &= \frac{\hat{\rho}_{k+1} - \sum_{j=1}^k \hat{\phi}_{kj} \hat{\rho}_{k+1-j}}{\sum_{j=1}^k \hat{\phi}_{kj} \hat{\rho}_j},\end{aligned}\quad (12)$$

dengan nilai $\hat{\phi}_{k+1,j} = \hat{\phi}_{kj} - \hat{\phi}_{k+1,k+1} \hat{\phi}_{k,k+1-j}$, $j = 1, 2, \dots, k$. Karakteristik plot ACF dan PACF untuk mengidentifikasi orde pada model ARIMA dapat dilihat pada Tabel 2.2.

Tabel 2.2 Pola Plot ACF dan PACF untuk Model Non Musiman

Model	ACF	PACF
AR(p)	Turun cepat (dies down)	Terpotong (cuts off) setelah lag p
MA(q)	Terpotong (cuts off) setelah lag q	Turun cepat (dies down)
ARMA(p,q)	Turun cepat (dies down)	Turun cepat (dies down)

2. Estimasi Parameter

Setelah didapatkan beberapa kemungkinan orde ARIMA(p, d, q), tahap selanjutnya adalah melakukan estimasi parameter. Estimasi parameter ini bertujuan untuk mendapatkan nilai dari setiap parameter yang terdapat di dalam model ARIMA. Metode *Moment Estimator*, *Least Square Estimator*, dan *Maximum Likelihood Estimator* (Cryer & Chan, 2008) merupakan beberapa metode yang biasa digunakan untuk melakukan estimasi parameter. Namun dari beberapa metode tersebut metode *Maximum Likelihood Estimator* (MLE) merupakan metode yang banyak digunakan karena memiliki beberapa kelebihan jika dibandingkan dengan metode yang lainnya. Menggunakan metode MLE ini semua informasi pada data digunakan dan tidak hanya terbatas pada momen pertama atau momen kedua saja. Metode MLE ini akan menggunakan fungsi kepadatan peluang gabungan seperti ditunjukkan oleh persamaan (2.15) dimana $\mathbf{a} = (a_1, a_2, \dots, a_T)'$ dan $a_t \sim N(0, \sigma_a^2)$:

$$P(\mathbf{a} | \phi, \mu, \theta, \sigma_a^2) = (2\pi\sigma_a^2)^{-\frac{T}{2}} \exp\left(-\frac{1}{2\sigma_a^2} \sum_{t=1}^T a_t^2\right) \quad (13)$$

Jika $a_t = \theta_1 a_{t-1} + \dots + \theta_q a_{t-q} + Y_t - \phi_1 Y_{t-1} - \dots - \phi_p Y_{t-p}$ dan $\mathbf{Y} = (Y_1, Y_2, \dots, Y_T)'$ serta diasumsikan bahwa kondisi awal untuk $\mathbf{Y}_* = (Y_{1-p}, \dots, Y_{-1}, Y_0)'$ dan $\mathbf{a}_* = (a_{1-q}, \dots, a_{-1}, a_0)'$, maka didapat fungsi log-likelihood dari persamaan (2.15) seperti pada persamaan (2.16) :

$$\ln L_*(\phi, \mu, \theta, \sigma_a^2) = -\frac{T}{2} \ln 2\pi \sigma_a^2 - \frac{S_*(\phi, \mu, \theta)}{2\sigma_a^2} \quad (14)$$

dimana $S_*(\phi, \mu, \theta) = \sum_{t=1}^T a_t^2(\phi, \mu, \theta | \mathbf{Y}_*, \mathbf{a}_*, \mathbf{Y})$ merupakan fungsi *conditional sum of square*. Nilai dari $\hat{\phi}$, $\hat{\mu}$, dan $\hat{\theta}$ akan memaksimumkan persamaan (2.16) sehingga disebut *conditional maximum likelihood estimators*. Deret $\{Y_t\}_{t=1}^T$ diasumsikan stasioner dan a_t diasumsikan *white noise*, sehingga nilai Y_t dapat diganti dengan rata-rata \bar{Y} dan nilai a_t diganti dengan nilai ekspektasinya yaitu 0 sehingga $S_*(\phi, \mu, \theta)$ dapat ditulis menjadi :

$$S_*(\phi, \mu, \theta) = \sum_{t=p+1}^T a_t^2(\phi, \mu, \theta | \mathbf{Y}) \quad (15)$$

Kemudian setelah didapatkan nilai estimasi parameter $\hat{\phi}$, $\hat{\mu}$, dan $\hat{\theta}$, nilai σ_a^2 dapat dihitung menggunakan persamaan 20.

$$\hat{\sigma}_a^2 = \frac{S_*(\hat{\phi}, \hat{\mu}, \hat{\theta})}{T - (2p + q + 1)} \quad (16)$$

Setelah didapatkan nilai estimasi dari parameter-parameter tadi, maka langkah selanjutnya adalah menguji signifikansi parameter-parameter tersebut. Jika $i=1, 2, \dots, p$, maka hipotesis yang digunakan untuk melakukan pengujian signifikansi parameter model AR adalah (Bowerman, O'Connell, & Koehler, 2004) :

$H_0 : \phi_i = 0$ (Parameter AR bernilai sama dengan nol atau tidak signifikan)

$H_1 : \phi_i \neq 0$ (Parameter AR bernilai tidak sama dengan nol atau signifikan)

$$\text{Statistik uji : } t_{hitung.i} = \frac{\hat{\phi}_i}{SE(\phi_i)}$$

H_0 akan ditolak apabila nilai statistik uji $|t_{hitung.i}| > t_{\alpha/2, (T-n_p)}$ atau nilai $p\text{-value} < \alpha$ dimana n_p adalah banyaknya parameter AR pada model, yaitu $n_p = p + 1$ apabila ada *intercept* dan $n_p = p$ apabila tidak ada *intercept*. Sedangkan jika $j=1, 2, \dots, q$, hipotesis yang digunakan untuk melakukan pengujian signifikansi model MA adalah :

$H_0 : \theta_j = 0$ (Parameter MA bernilai sama dengan nol atau tidak signifikan)

$H_1 : \theta_j \neq 0$ (Parameter MA bernilai tidak sama dengan nol atau signifikan)

$$\text{Statistik uji : } t_{hitung.j} = \frac{\hat{\theta}_j}{SE(\theta_j)}.$$

H_0 akan ditolak apabila nilai statistik uji $|t_{hitung.j}| > t_{\alpha/2, (T-n_q)}$ atau nilai $p\text{-value} < \alpha$ dimana n_q adalah banyaknya parameter MA pada model, yaitu $n_q = q + 1$ apabila ada *intercept* dan $n_q = q$ apabila tidak ada *intercept*.

3. Cek Diagnosa

Langkah berikutnya adalah mengecek kesesuaian model melalui cek diagnosa. Seperti yang telah disinggung sebelumnya, model ARIMA (p, d, q) harus memenuhi asumsi residual a_t yang *white noise* dan berdistribusi normal. Digunakan uji Ljung-Box untuk mengetahui apakah a_t merupakan proses yang identik dan independen. Jika K merupakan panjang *lag* yang diuji, hipotesis yang digunakan untuk uji Ljung-Box adalah:

$H_0 : \rho_1 = \rho_2 = \dots = \rho_k = \dots = \rho_K = 0$ (Antar residual tidak ada korelasi atau model telah independen)

H_1 : minimal ada satu nilai $\rho_k \neq 0$ dimana $k=1,2,\dots,K$ (Ada korelasi dalam residual atau model belum independen)

Persamaan statistik uji Q adalah:

$$Q = T(T+2) \sum_{k=1}^K \frac{\hat{\rho}_k^2}{T-k} \quad (17)$$

Nilai statistik uji Q tersebut diketahui mengikuti distribusi *chi-square* dengan derajat bebas $K-p-q$ dimana nilai p dan q merupakan orde dari model $ARIMA(p,d,q)$. H_0 akan ditolak apabila nilai $Q > \chi^2_{K-p-q,\alpha}$ tabel atau $p\text{-value} < \alpha$ dimana nilai p adalah banyaknya parameter AR pada model dan q adalah banyaknya parameter MA pada model.

Uji asumsi *white-noise* menggunakan uji Ljung-Box biasanya masih menyisakan beberapa *lag residual* yang signifikan yang menandakan bahwa *residual* masih belum sepenuhnya acak. Karena itu, dalam beberapa penelitian juga digunakan *plot ACF residual* untuk menguji asumsi *white-noise* (Ramasubramanian, 2007). Ketika sudah tidak ada nilai autokorelasi yang melebihi garis batas $\pm z_{\alpha/2} / \sqrt{N}$ maka dapat disimpulkan bahwa nilai *residual* yang didapatkan sudah memenuhi asumsi *white-noise*.

Untuk menguji apakah residual telah memenuhi asumsi berdistribusi normal atau tidak maka digunakan metode *Kolmogorov-Smirnov*. Konsep dari metode ini adalah membandingkan fungsi distribusi empiris atau $F(a_t)$ dengan fungsi distribusi hipotesis atau $F_0(a_t)$ dan dalam hal ini adalah distribusi normal. Hipotesis yang digunakan pada pengujian ini adalah sebagai berikut :

H_0 : $F(a_t) = F_0(a_t)$ (Residual mengikuti distribusi normal)

H_1 : $F(a_t) \neq F_0(a_t)$ (Residual tidak mengikuti distribusi normal)

dengan statistik uji:

$$D = \text{Sup} |F(a_t) - F_0(a_t)|. \quad (18)$$

Keterangan:

$F(a_t)$ = fungsi distribusi frekuensi kumulatif *residual*

$F_0(a_t)$ = fungsi distribusi frekuensi kumulatif distribusi normal

Sup = nilai maksimum dari semua hasil $|F(a_t) - F_0(a_t)|$

Keputusan untuk menolak H_0 dilakukan jika nilai D lebih besar dari nilai tabel *Kolmogorov-Smirnov* yaitu $d_{T,\alpha}$ dimana T adalah banyaknya residual yang diuji dan α adalah taraf signifikansi yang digunakan. (O'Connor & Kleyner, 2012).

2.3 Evaluasi Model

Evaluasi model dan pemilihan model terbaik dilakukan menggunakan nilai *root mean square error* (RMSE). Setelah dilakukan pemodelan, maka tiap-tiap model dihitung nilai RMSE-nya dan kemudian dilakukan perbandingan antar metode. Model terbaik merupakan model yang menghasilkan RMSE paling kecil. RMSE *out-sample* dapat diperoleh dengan menggunakan rumus pada persamaan (19) (Wei, 2006) dimana N merupakan banyaknya data *out-sample*.

$$RMSE_{out} = \sqrt{MSE_{out}} = \sqrt{\frac{1}{N} \sum_{t=1}^N (Y_t - \hat{Y}_t)^2} \quad (19)$$

2.4 Jenis Outlier dan Metode Pendeteksian Outlier

Outlier adalah data pengamatan yang tidak konsisten pada deretnya. Efek kejadian tersebut dapat dihitung dengan model intervensi jika waktu dan penyebab diketahui. Ada empat macam jenis *outlier* yaitu *Innovational Outlier* (IO), *Additive Outlier* (AO), *Temporary Change* (TC), dan *Level Shift* (LS). Deteksi *outlier* pertama kali dikemukakan Fox (1972) yang memperkenalkan *outlier* tipe 1 atau *additive outliers* (AO) dan tipe 2 atau *innovation outliers* (IO) (Wei W. , 2006). Pada data *time series*, *outlier* perlu diatasi supaya karakteristik data *time series* menjadi lebih baik sehingga menghasilkan peramalan, model dan estimasi yang lebih baik dan sempurna (Taylor & McSharry, 2008).

2.4.1 Additive Outlier (AO)

Additive outlier adalah kejadian yang mempunyai efek pada data *time series* hanya pada satu periode saja. Bentuk umum sebuah *Additive Outliers* (AO) dalam proses ARMA diuraikan sebagai berikut:

$$\begin{aligned}
 Z_t &= \begin{cases} X_t & t \neq T \\ X_t + \omega & t = T \end{cases} \\
 &= X_t + \omega_{AO} I_t^{(T)} \\
 &= \frac{\theta(B)}{\phi(B)} a_t + \omega_{AO} I_t^{(T)} \quad (20)
 \end{aligned}$$

dengan

$$I_t^{(T)} = \begin{cases} 1 & t = T \\ 0 & t \neq T \end{cases}$$

adalah variabel indikator yang mewakili ada atau tidak adanya *outlier* pada waktu T.

2.4.2 Innovational Outlier (IO)

Innovational outliers adalah kejadian yang efeknya mengikuti proses ARMA. Bentuk umum sebuah *innovational outliers* didefinisikan sebagai berikut:

$$Z_t = X_t + \frac{\theta(B)}{\phi(B)} \omega_{IO} I_t^{(T)} = \frac{\theta(B)}{\phi(B)} (a_t + \omega_{IO} I_t^{(T)}) \quad (21)$$

Dari persamaan-persamaan tersebut, dapat disimpulkan bahwa *additive outlier* (AO) hanya mempengaruhi pengamatan ke-T, sedangkan *innovational outlier* (IO) mempengaruhi semua pengamatan Z_T, Z_{T+1}, \dots , melebihi waktu T sepanjang memori sistem yang dijelaskan oleh $\frac{\theta(B)}{\phi(B)}$.

Secara umum, sebuah data *time series* bisa saja mengandung beberapa *outlier*, misalnya k buah *outlier* dengan tipe yang

berbeda. Sehingga model umum *outlier* dapat ditulis sebagai berikut:

$$Z_t = \sum_{j=1}^k \omega_j v_j(B) I_t^{(T_j)} + X_t, \quad (22)$$

dimana $X_t = (\theta(B) / \phi(B)) a_t, v_j(B) = 1$, untuk AO dan $v_j(B) = \theta(B) / \phi(B)$ untuk IO pada waktu $t = T_j$. (Wei W. , 2006)

2.4.3 Temporary Change (TC)

Sedangkan TC adalah suatu kejadian dimana *outlier* menghasilkan efek awal sebesar ω pada waktu t , kemudian secara perlahan sesuai dengan besarnya δ . Model TC dapat dituliskan sebagai berikut:

$$\begin{aligned} Z_t &= X_t + \frac{1}{(1 - \delta B)} \omega_{TC} I_t^{(T)} \\ &= \frac{\theta(B)}{\phi(B)} a_t + \frac{1}{(1 - \delta B)} \omega_{TC} I_t^{(T)} \end{aligned} \quad (23)$$

Pada saat $\delta = 0$ maka TC akan menjadi kasus *additive outlier*, sedangkan pada saat $\delta = 1$ maka TC akan menjadi kasus *level shift*.

2.4.4 Level Shift (LS)

Selain dua tipe *outlier* tersebut, masih ada dua tipe *outlier* lain yang sering dibahas dalam analisis *time series*, yaitu *Level Shift* (LS) dan *Temporary Change* (TC). Suatu LS adalah kejadian yang mempengaruhi deret pada satu waktu tertentu yang memberikan suatu perubahan tiba-tiba dan permanen. Model *outlier* LS dinyatakan sebagai:

$$\begin{aligned}
Z_t &= X_t + \frac{1}{(1-B)} \omega_{LS} I_t^{(T)} \\
&= \frac{\theta(B)}{\phi(B)} + \frac{1}{(1-B)} \omega_{LS} I_t^{(T)} \\
&= \frac{\theta(B)}{\phi(B)} + \omega_{LS} S_t^{(T)}
\end{aligned} \tag{24}$$

dengan

$$S_t^{(T)} = \begin{cases} 1, & t \geq T \\ 0, & t < T \end{cases}$$

2.4.5 Metode Pendeteksian *Outlier* Chen dan Liu (1993)

Misalkan deret Y_t dikenakan intervensi sampai m pada titik t_1, t_2, \dots, t_m sehingga menghasilkan berbagai jenis *outlier*. Model untuk Y_t^* dapat dinyatakan sebagai berikut.

$$Y_t^* = \sum_{j=1}^m \omega_j L_j(B) I_t(t_j) + \frac{\theta(B)}{\phi(B)\alpha(B)} a_t, \tag{25}$$

dimana $L_j(B) = \frac{\theta(B)}{\phi(B)\alpha(B)}$ untuk IO, $L_j(B) = 1$ untuk AO,

$L_j(B) = \frac{1}{(1-B)}$ untuk LS, dan $L_j(B) = \frac{1}{(1-\delta B)}$ untuk TC pada

saat $t = t_j$. Tanpa membedakan notasi dari estimasi parameter maupun parameter yang sebenarnya, residual dapat dinyatakan sebagai:

$$\hat{e}_t = \sum_{j=1}^m \omega_j \pi(B) L_j(B) I_t(t_j) + a_t, \tag{26}$$

ketika model dasar sudah ditentukan dengan benar namun efek *outlier* tidak dipertimbangkan. Persamaan (25) dan (26) adalah dasar dari prosedur yang diusulkan Chen dan Liu (1993). Jika efek

outlier dan lokasinya diketahui, maka efek outlier dapat disesuaikan berdasarkan Persamaan (25) dan selanjutnya mengestimasi parameter model. Di sisi lain, ketika parameter model diketahui outlier dapat diidentifikasi dan diperkirakan efeknya berdasarkan Persamaan (26). Hal ini sulit, namun bukan tidak mungkin untuk dapat mencapai tujuan dalam satu langkah sekaligus. Sehingga Chen dan Liu (1993) mengembangkan prosedur iterasi yang terdiri dari tiga tahap utama. Dalam Tahap I semua observasi yang berpotensi sebagai outlier yaitu t_j dan

$L_j(B)$ diidentifikasi berdasarkan estimasi awal parameter model.

Dalam Tahap II estimasi gabungan dari parameter model dan efek outlier diperoleh menggunakan informasi akumulasi outlier dari Tahap I. Dalam Tahap III outlier t_j dan $L_j(B)$ diidentifikasi dan efeknya diestimasi lagi berdasarkan estimasi yang paling sedikit terkontaminasi dari parameter model yang diperoleh di Tahap II.

1. Tahap I : Estimasi Parameter Awal dan Pendeteksian Outlier

- I.1 Menghitung *maximum likelihood estimates* dari parameter model asal atau dari deret yang disesuaikan lalu mendapatkan residual. Untuk iterasi pertama, deret asal digunakan untuk memulai prosedur. Kemudian setelah iterasi pertama, deret disesuaikan.

Pendeteksian Outlier Loop Dalam Untuk Estimasi Parameter Model Tetap

- I.2 Menghitung $\hat{\tau}_{IO}(t)$, $\hat{\tau}_{AO}(t)$, $\hat{\tau}_{LS}(t)$, $\hat{\tau}_{TC}(t)$ untuk $t = 1, \dots, n$ dengan rumus sebagai berikut

$$\begin{aligned}
\hat{\tau}_{IO}(t_1) &= \frac{\hat{\omega}_{IO}(t_1)}{\hat{\sigma}_a} \\
\hat{\tau}_{AO}(t) &= \frac{\hat{\omega}_{AO}(t_1)}{\hat{\sigma}_a} \left(\sum_{t=t_1}^n x_{2t}^2 \right)^{1/2} \\
\hat{\tau}_{LS}(t) &= \frac{\hat{\omega}_{LS}(t_1)}{\hat{\sigma}_a} \left(\sum_{t=t_1}^n x_{3t}^2 \right)^{1/2} \\
\hat{\tau}_{TC}(t) &= \frac{\hat{\omega}_{TC}(t_1)}{\hat{\sigma}_a} \left(\sum_{t=t_1}^n x_{4t}^2 \right)^{1/2}
\end{aligned} \tag{27}$$

dengan menggunakan residual yang didapatkan dari I.1 dan menghitung $\eta_t = \max \left\{ |\hat{\tau}_{IO}(t)|, |\hat{\tau}_{AO}(t)|, |\hat{\tau}_{LS}(t)|, |\hat{\tau}_{TC}(t)| \right\}$.

Jika $\max_t \eta_t = |\hat{\tau}_{tp}(t_1)| > C$, dimana C merupakan nilai kritis yang sudah ditentukan sebelumnya, maka ada kemungkinan terdapat outlier dengan jenis tp di t_1 , tp bisa saja outlier dengan jenis IO, AO, LS maupun TC.

- I.3 Jika tidak ditemukan outlier, maka langsung ke langkah I.4. Namun jika ditemukan outlier, maka efek outlier dari residual dan observasi berdasarkan tipenya dihapus. Kemudian kembali ke langkah I.2 untuk memeriksa apakah ada outlier lagi dapat yang ditemukan.
- I.4 Jika tidak ada outlier yang ditemukan pada iterasi pertama dari loop dalam, maka proses dihentikan. Artinya deret pengamatan bebas dari efek outlier. Jika outlier ditemukan di loop dalam pada estimasi parameter yang diberikan, maka kembali ke langkah I.1 untuk memperbarui estimasi parameter. Jika jumlah outlier dalam semua loop dalam lebih besar dari 0 dan tidak ada outlier yang terdeteksi lagi di loop dalam, maka langsung menuju langkah II.1.

2. Tahap II : Estimasi Bersama Efek Outlier dan Parameter Model

- II.1 Misalkan m titik waktu t_1, t_2, \dots, t_m diidentifikasi berbagai tipe outlier yang mungkin. Efek outlier ω_j dapat diestimasi secara bersama dengan menggunakan model regresi berganda yang diuraikan dalam persamaan (26), dimana e_t dianggap sebagai variabel output dan $L_j(B)I_t(t_j)$ adalah variabel input.
- II.2 Menghitung statistik uji $\hat{\tau}$ dari estimasi ω_j , dimana $\hat{\tau}_j = \hat{\omega}_j / \text{std}(\hat{\omega}_j)$, $j = 1, \dots, m$. Jika $\min_j |\hat{\tau}_j| = \hat{\tau}_p \leq C$, dimana C adalah nilai kritis yang sama yang digunakan pada langkah I.2, maka outlier pada titik waktu ke- t_p dihapus dari deret dan kembali ke langkah II.1 dengan jumlah outlier yang tersisa adalah $m-1$. Namun jika tidak, langsung menuju langkah II.3.
- II.3 Mendapatkan deret yang disesuaikan dengan menghapus efek outlier menggunakan estimasi terbaru dari ω_j pada langkah II.1. Dengan kata lain, hanya outlier yang signifikan berdasarkan iterasi pada langkah II.1 dan II.2 saja yang dihapus.
- II.4 Menghitung *maximum likelihood estimates* dari parameter model berdasarkan deret yang sudah disesuaikan yang didapatkan pada langkah II.3. Jika perubahan relatif dari standard error residual dari estimasi sebelumnya lebih besar dari ε maka kembali ke langkah II.1 untuk iterasi selanjutnya. Jika tidak maka dilanjutkan ke langkah III.1. Toleransi ε adalah konstanta yang sudah ditentukan sebelumnya oleh peneliti sebagai cara untuk mengontrol akurasi estimasi parameter.

3. Tahap III : Deteksi Outlier Berdasarkan Estimasi Parameter Akhir

- III.1 Menghitung residual dengan memilih deret asli berdasarkan estimasi parameter yang diperoleh pada langkah II.4.
- III.2 Menggunakan residual yang diperoleh dari langkah III.1 dan iterasi melalui Tahap I dan II dengan modifikasi (a) estimasi parameter yang digunakan dalam loop dalam Tahap I adalah sama dengan yang diperoleh pada langkah II.4 dan (b) langkah II.3 dan II.4 dihilangkan dalam Tahap II. Estimasi ω_j dari iterasi terakhir pada langkah II.1 adalah estimasi akhir dari efek outlier yang terdeteksi.

2.5 Metode Window Time

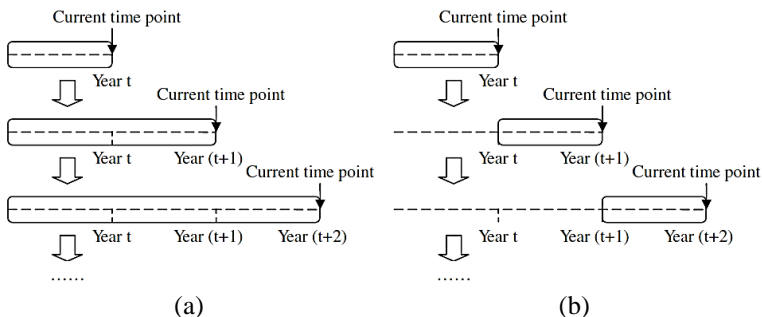
Istilah *window time* berkaitan erat dengan konsep *drift* (Sun & Li, 2011). Terdapat lima macam jenis pembagian jendela yang digunakan dalam pemodelan yaitu *full memory* dan *no memory*, *fixed size* dan *adaptable size*, serta *batch selection*. Masing-masing ilustrasi akan dijelaskan dalam sub bab berikut.

2.5.1 Window Time Full Memory dan No Memory

Metode *window time full memory* mengasumsikan bahwa mengabaikan *window time* sebelumnya tidak diperlukan dalam pemodelan. Model dihasilkan dari semua *window time* pada interval sebelumnya dan observasi terbaru ditambahkan ke *window time* yang tergabung dalam interval. Sementara itu, tidak ada *window time* lama yang dihapus dari lebar jendela. Seperti yang ditunjukkan pada Gambar 2.4 (a), ukuran jendela menjadi semakin besar dengan titik waktu beralih dari tahun (t) ke tahun ($t + 1$) dan kemudian dari tahun ($t + 1$) untuk tahun ($t + 2$). Namun, kelemahan dari metode ini adalah *full memory window time* tidak bisa beradaptasi dengan konsep baru dengan baik karena model baru mewarisi baik konsep lama maupun konsep baru. Oleh karena itu, *full memory window time* hanya cocok diterapkan pada konsep dimana periode lama dan periode baru memiliki informasi yang

sama-sama penting. Kelemahan lain dari metode ini adalah bahwa *full memory window time* secara bertahap akan menjadi terlalu besar terutama ketika jumlah data yang ada di setiap periode semakin bertambah seiring dengan berjalannya waktu.

Acuan *no memory window time* adalah menggunakan jendela dengan ukuran yang tetap dari satu kumpulan data. Metode ini mengasumsikan bahwa kumpulan data pembentuk tidak berhubungan dengan konsep data saat ini, dan model baru harus dibangun dari kumpulan data terbaru pada setiap titik waktu yang baru pula dengan mengabaikan semua informasi lama. Gambar 2.4 (b) menunjukkan gagasan *no memory window time*. Ide ini sepenuhnya berlawanan dengan *full memory window time*. Kelemahan dari metode ini adalah bahwa model dibangun dari *no memory window time* sehingga tidak bisa dilakukan generalisasi untuk data *training* dengan jumlah yang terbatas dalam periode waktu ketika konsep terus berjalan stabil.

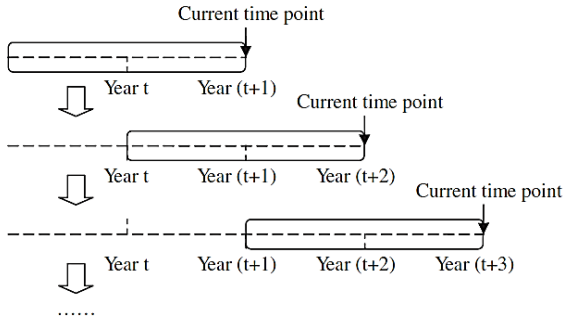


Gambar 2.4 (a) *Full Memory Time Window* dan (b) *No Memory Time Window* (Klinkenberg, 2004)

2.5.2 Window Time Fixed Size dan Adaptable Size

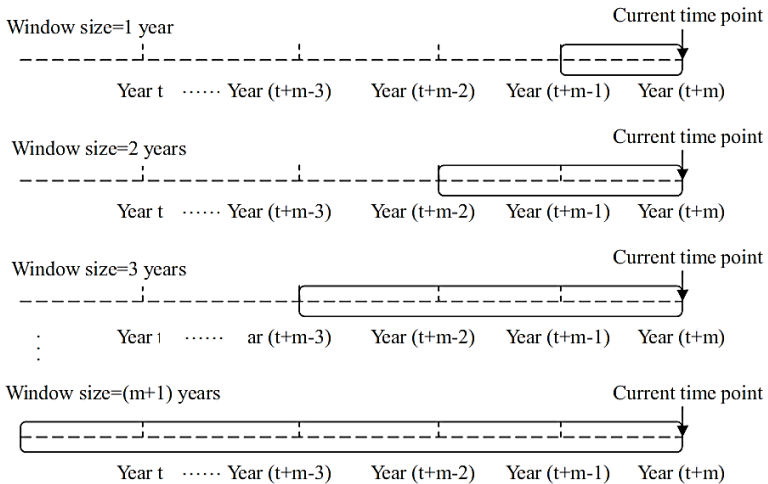
Permasalahan utama *fixed size window time* adalah bagaimana memilih ukuran jendela yang sesuai. *Window time* yang sempit memiliki kemampuan beradaptasi yang kuat dengan konsep drift, namun tidak dapat digeneralisasi saat konsep drift terlalu sederhana karena terbatasnya jumlah observasi. Sebaliknya, *window time* yang lebar dipastikan dapat digeneralisasi, namun

dalam beberapa kasus banyak informasi lama yang tidak cocok untuk kasus baru (Klinkenberg, 2004). Gambar 2.5 mengilustrasikan pemodelan pada dasar *window time* saat ukuran tetap yaitu 2 tahun.



Gambar 2.5 Fixed Size Time Window 2 tahun (Klinkenberg, 2004)

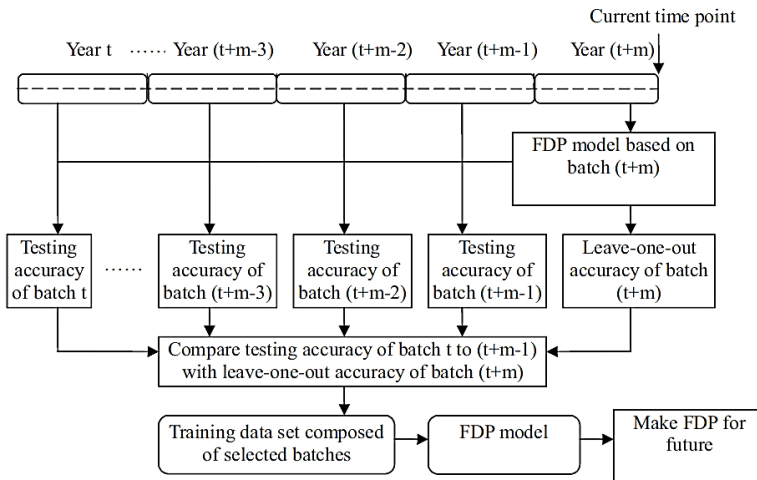
Untuk *adaptable size window time*, ukuran jendela disesuaikan oleh beberapa mekanisme. Widmer dan Kubat (1996) mengusulkan adaptif *window time* dengan heuristik, yaitu melibatkan beberapa parameter. Klinkenberg & Joachims (2000) menyajikan pendekatan untuk memilih ukuran jendela sehingga dapat meminimalkan kesalahan generalisasi pada kumpulan data terbaru. Misalkan titik waktu saat ini adalah tahun $(t + m)$, sehingga terdapat $(m + 1)$ window yang mungkin terjadi, seperti yang ditunjukkan pada Gambar 2.7. Dengan asumsi bahwa kumpulan data terbaru yaitu tahun $(t + m)$ adalah yang paling mirip dengan prediksi yang akan datang.



Gambar 2.6 *Adaptable Size Time Window* (Klinkenberg, 2004)

2.5.3 Metode *Batch Selection*

Klinkenberg (2004) mengusulkan metode pemilihan kumpulan data yang berbeda dari metode *windowing* tradisional karena kasus yang dipilih tidak mencakup beberapa berdekatan data terbaru. Sebaliknya, metode ini memilih kumpulan data yang mirip dengan kumpulan data terbaru dengan tidak memperhatikan lokasi pemotongan *window*. Dasar ide metode ini ditampilkan dalam Gambar 2.7. Pertama, model dipelajari dari kumpulan data terbaru dari $(t + m)$. Meskipun model ini tidak cukup baik untuk memprediksi masa depan dalam banyak kasus, namun metode tersebut merupakan yang paling banyak dipakai dan diyakini dapat mewakili konsep saat ini. Oleh karena itu, model ini dapat digunakan untuk menilai mana kumpulan data lama yang dihasilkan dari konsep yang mirip dengan kumpulan data terbaru dengan membandingkan akurasi pengujian model pada kumpulan data lama.



Gambar 2.7 *Batch Selection* (Klinkenberg, 2004)

2.6 Uji ANOVA

Uji asumsi Anova dibagi menjadi 2 yaitu uji kenormalan data dan uji homogenitas data.

2.6.1 Uji Asumsi Kenormalan

Uji asumsi kenormalan bertujuan untuk mengetahui apakah residual/error terdistribusi secara normal dengan IDN $(0, \sigma^2)$. Uji asumsi kenormalan dapat dilakukan dengan 2 cara yaitu visual dan analitis. Data dikatakan terdistribusi normal secara visual apabila residual plotnya menyerupai garis lurus.

Langkah-langkah uji kenormalan data secara analitis adalah sebagai berikut.

Hipotesis:

H_0 : Residual plot terdistribusi normal

H_1 : Residual plot terdistribusi tidak normal

Pengambilan keputusan:

Jika nilai $p > \alpha$, maka H_0 diterima

Jika nilai $p < \alpha$, maka H_0 ditolak

2.6.2 Uji Asumsi Homogenitas

Uji homogenitas data bertujuan untuk mengetahui apakah kombinasi perlakuan pada eksperimen memiliki varian yang sama atau tidak. Jenis uji homogenitas ada bermacam-macam antara lain uji Barlett untuk faktor dengan tiga level dan uji F untuk faktor dengan dua level.

Hipotesis:

$H_0: \sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \dots = \sigma_k^2$ (Varian homogen)

$H_1: \text{Ada } \sigma_i^2 \neq \sigma_j^2 \text{ dengan } i \neq j$ (Varian tidak homogen)

Pengambilan keputusan:

Jika nilai $p > \alpha$, maka H_0 diterima

Jika nilai $p < \alpha$, maka H_0 ditolak

2.6.3 Uji ANOVA

Analysis of variance atau ANOVA merupakan salah satu uji parametrik yang berfungsi untuk membedakan nilai rata-rata lebih dari dua kelompok data dengan cara membandingkan variansinya (Ghozali, 2009). Prinsip uji Anova adalah melakukan analisis variabilitas data menjadi dua sumber variasi yaitu variasi di dalam kelompok (within) dan variasi antar kelompok (between). Bila variasi within dan between sama (nilai perbandingan kedua varian mendekati angka satu), berarti nilai mean yang dibandingkan tidak ada perbedaan. Sebaliknya bila variasi antar kelompok lebih besar dari variasi didalam kelompok, nilai mean yang dibandingkan menunjukkan adanya perbedaan.

Uji ANOVA dapat dibagi menjadi 2 jenis berdasarkan jumlah variabel yang diamati, yaitu *One Way ANOVA* dan *Two Way ANOVA*. *One Way ANOVA* digunakan bila ada satu variabel yang ingin diamati, sedangkan *Two Way ANOVA* digunakan apabila terdapat dua variabel yang ingin diamati. Sedangkan untuk menganalisis data dengan faktor yang lebih banyak dapat menggunakan *Multi Way ANOVA*. Untuk memudahkan perhitungan ANOVA, maka dapat digunakan tabel ANOVA yang ditunjukkan oleh Tabel 2.3 berikut.

Tabel 2.3 Tabel *Multi Way ANOVA*

Source of Variation	df	SS	MS	F
Faktor A	a-1	$\sum_{i=1}^a n_i (\bar{y}_i - \bar{y}_{..})^2$	$\frac{SSA}{(a-1)}$	$\frac{MSA}{MSE}$
Faktor B	b-1	$\sum_{j=1}^b n_j (\bar{y}_j - \bar{y}_{..})^2$	$\frac{SSB}{(b-1)}$	$\frac{MSB}{MSE}$
Faktor C	c-1	$\sum_{k=1}^c n_k (\bar{y}_k - \bar{y}_{..})^2$	$\frac{SSC}{(c-1)}$	$\frac{MSC}{MSE}$
Faktor D	d-1	$\sum_{l=1}^d n_l (\bar{y}_l - \bar{y}_{..})^2$	$\frac{SSD}{(d-1)}$	$\frac{MSD}{MSE}$
Error	(a-1) (b-1) (c-1) (d-1)	SST-SSA- SSB-SSC- SSD	$\frac{SSE}{(a-1)(b-1)(c-1)(d-1)}$	
Total	N-1	$\sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^c \sum_{l=1}^d (y_{ijkl} - \bar{y}_{..})^2$		

Uji *ANOVA* dapat digunakan untuk menyelidiki apakah ada pengaruh faktor terhadap respon penelitian. Uji-uji yang dapat digunakan antara lain uji masing-masing faktor dan uji interaksi antar faktor.

BAB III METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan merupakan simulasi dari data deret waktu dengan model ARIMA (1,0,0) dengan $\phi=0.8, -0.8, 0.5$ dan -0.5 yang dibangkitkan menggunakan program RStudio Version 0.99.903. Kemudian pada masing-masing data tersebut disisipkan *outlier* tunggal di dalamnya. Jenis *outlier* yang disisipkan adalah AO, IO, TC dan LS. Panjang data yang disimulasikan sebanyak 5000 data, *critical value* yang digunakan sebesar 4, $\mathcal{S}=0.7$ dan besarnya *outlier* ditentukan sebesar 4.

3.2 Langkah Analisis

Langkah penelitian yang digunakan dalam analisis adalah sebagai berikut.

1. Membangkitkan data simulasi masing-masing 100 data dengan model ARIMA (1,0,0) dengan besar parameter yang ditentukan dan panjang data sebanyak 5000 dengan residual yang memenuhi IIDN (0,1).
2. Menghapus 100 data awal sehingga data yang akan digunakan dalam observasi sebanyak 4900 data.
3. Memvalidasi masing-masing model yang telah dibangkitkan apakah sesuai dengan model penelitian yang diinginkan.
4. Menambahkan efek outlier tunggal pada masing-masing model data. Empat jenis *outlier* yang disisipkan adalah AO, IO, TC dan LS. Masing-masing penyisipan *outlier* tersebut dikombinasi dengan lokasi *outlier* tersebut diletakkan yaitu di awal ($T = 1300$), tengah ($T = 2500$) dan akhir data ($T = 3700$). Sehingga terdapat 36 kombinasi yang dihasilkan dari 3 model, 4 jenis outlier dan 3 lokasi yang berbeda.
5. Membagi data menjadi 4800 data *in sampel* dan 100 data *out sampel*.

6. Mendeteksi *outlier* yang ada dalam data dengan kombinasi panjang data awal yang dideteksi sebanyak 100, 500 dan 1000. Serta mengkombinasikan lokasi *outlier* yaitu di awal, tengah dan akhir data.

Metode deteksi outlier sebelumnya yang dilakukan oleh Chen & Liu (1993) diilustrasikan dalam Gambar 3.1 sedangkan metode baru yang akan dilakukan untuk menentukan panjang optimal data deret waktu bebas *outlier* diilustrasikan dalam Gambar 3.2 dan 3.3 dan dijabarkan dalam prosedur sebagai berikut:

Jumlah data awal yang digunakan adalah sebanyak 100, 500 dan 1000 dengan panjang pergeseran sebesar 100 data.

- a. Memodelkan data in sampel keseluruhan
- b. Model yang didapatkan dari keseluruhan data in sampel tersebut digunakan untuk mendeteksi *outlier* pada 100 observasi in sample terakhir.
- c. Apabila *outlier* terdeteksi maka *outlier* tersebut dikeluarkan dari series sampai tidak ada *outlier* lagi.
- d. Setelah 100 observasi tersebut bersih dari *outlier* lalu dimodelkan.
- e. Model yang didapatkan dari 100 observasi terakhir tersebut digunakan untuk mendeteksi *outlier* pada 200 observasi in sample terakhir.
- f. Apabila *outlier* terdeteksi maka *outlier* tersebut dikeluarkan dari series sampai tidak ada *outlier* lagi. Proses terus berlanjut sampai data observasi habis dan bersih dari *outlier*.

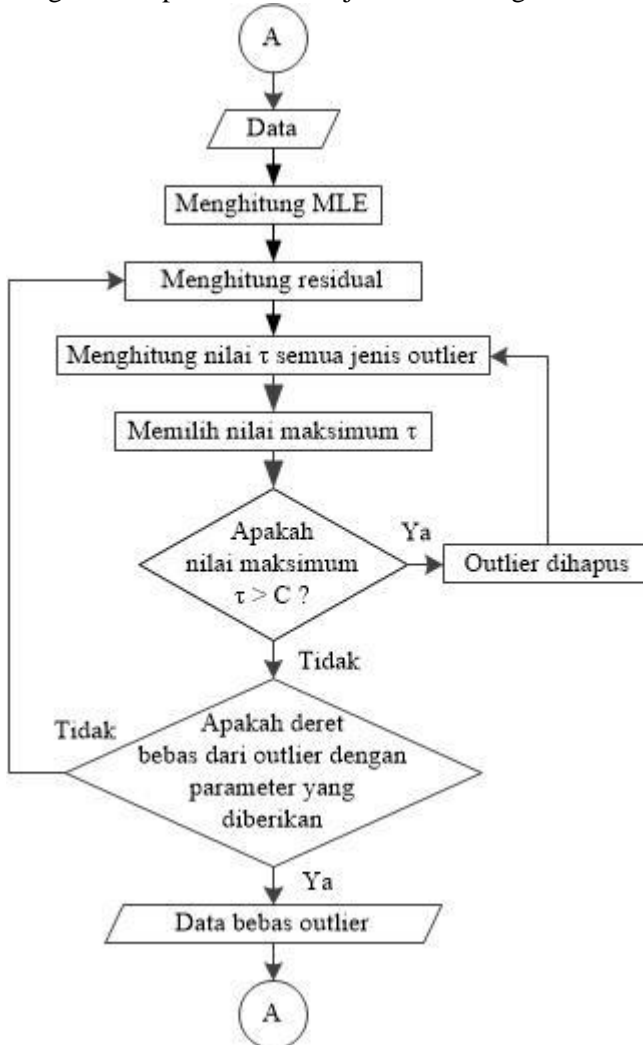
Dengan langkah-langkah yang sama dilakukan untuk panjang data awal yang diobservasi sebesar 500 dan 1000 yang terletak di awal dan tengah *series*.

7. Menghitung kesalahan pendeteksian *outlier* pada masing-masing data.
8. Melakukan pengujian ANOVA multi way pada persentase kesalahan pendeteksian *outlier* pada masing-masing model.

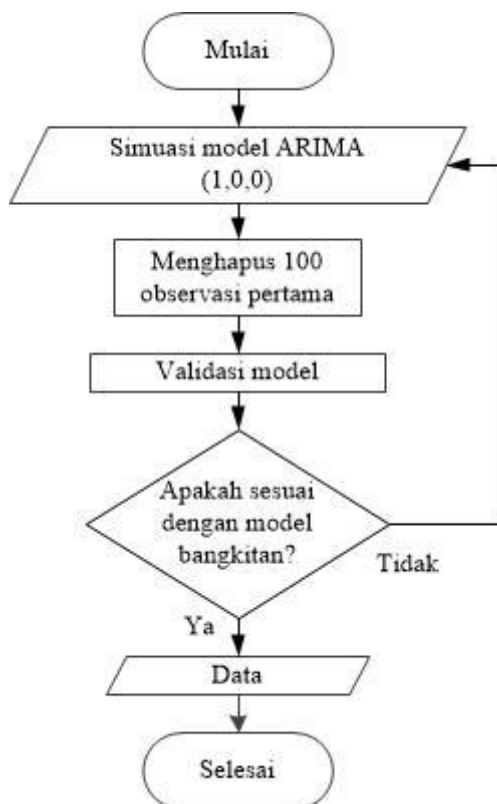
9. Mendapatkan panjang optimal data yang dibutuhkan untuk memprediksi suatu data deret waktu dengan model AR(1) yang bebas *outlier*.
10. Menghitung RMSE out sample dari tiga cara yaitu: (1) prediksi tanpa melakukan deteksi *outlier* pada data, (2) prediksi dengan melakukan deteksi *outlier* di keseluruhan data, dan (3) prediksi dengan melakukan deteksi *outlier* dan *window time*.

3.3 Diagram Alir

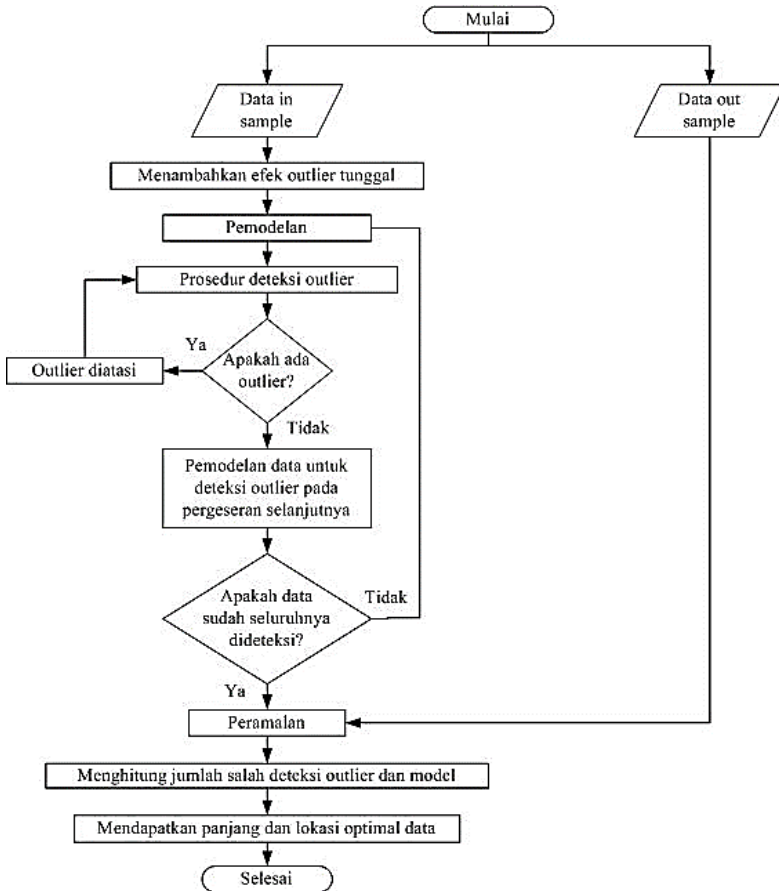
Diagram alir penelitian ditunjukkan dalam gambar berikut.



Gambar 3.1 Diagram Alir Langkah Deteksi *Outlier* Chen & Liu (1993)



Gambar 3.2 Diagram Alir Langkah Simulasi Data



Gambar 3.3 Diagram Alir Langkah Penyisipan *Outlier* dan Proses Deteksi *Outlier*

BAB IV ANALISIS DAN PEMBAHASAN

4.1 Data Simulasi

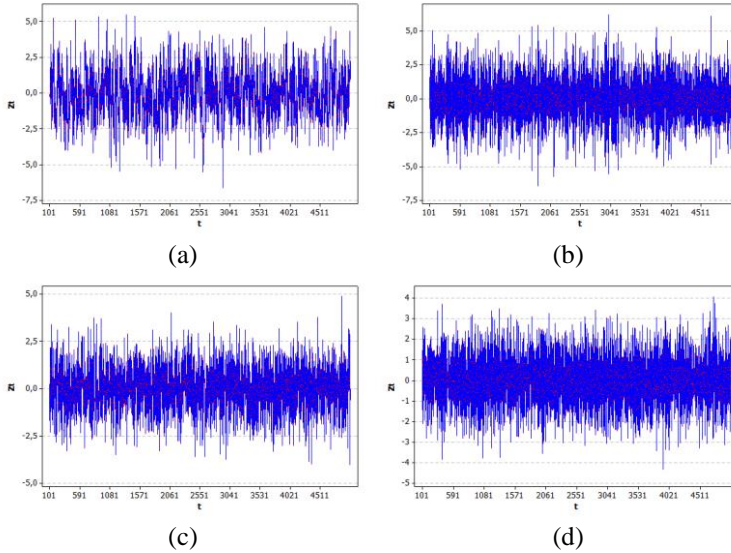
Data simulasi dibangkitkan dari model ARIMA (1,0,0) dengan 4 nilai parameter yang berbeda-beda baik parameter yang bernilai positif maupun negatif. Banyaknya deret yang dibangkitkan adalah 5000 observasi dan banyaknya perulangan yang dibangkitkan dalam setiap model dengan parameter berbeda adalah 100 kali. Kemudian, data simulasi tersebut disisipkan *outlier* dengan jenis *Additional Outlier* (AO), *Innovational Outlier* (IO), *Temporary Change* (TC) atau *Level Shift* (LS) di lokasi yang berbeda-beda. *Critical value* yang digunakan sebesar 4, begitu juga dengan besaran *outlier* ditentukan sebesar 4. Proses simulasi data dilakukan menggunakan bantuan program RStudio Version 0.99.903. Berikut merupakan data dengan model ARIMA (1,0,0) yang dibangkitkan dengan 4 variasi parameter.

Tabel 4.1 Empat Model yang Digunakan Dalam Simulasi

No.	Model
1.	$Z_t = -0,8Z_{t-1} + a_t, a_t \sim N(0,1)$
2.	$Z_t = 0,8Z_{t-1} + a_t, a_t \sim N(0,1)$
3.	$Z_t = -0,5Z_{t-1} + a_t, a_t \sim N(0,1)$
4.	$Z_t = 0,5Z_{t-1} + a_t, a_t \sim N(0,1)$

Setiap model ARIMA (1,0,0) dengan parameter yang sudah ditentukan tersebut dibangkitkan sebanyak 100 kali perulangan supaya memberikan hasil yang terbaik. Pada 100 observasi pertama dari 5000 data bangkitan di setiap data bangkitan dihapus karena pada awal proses bangkitan belum menghasilkan model ARIMA (1,0,0) yang konvergen. Semua data bangkitan juga tidak seluruhnya dipakai dalam tahap penelitian selanjutnya. Setiap data harus dilakukan validasi terlebih dahulu untuk memastikan bahwa data bangkitan mengikuti model yang diinginkan. Sehingga pada akhirnya dipilih 100 data untuk masing-masing model yang benar-

benar valid mengikuti model ARIMA(1,0,0) dengan parameter yang sesuai. Berikut merupakan *time series plot* dari data bangkitan setiap model.



Gambar 4.1 Time Series Plot Model Simulasi Perulangan Pertama (a)

$$Z_t = -0,8Z_{t-1} + a_t \quad (b) \quad Z_t = 0,8Z_{t-1} + a_t \quad (c) \quad Z_t = -0,5Z_{t-1} + a_t \quad (d)$$

$$Z_t = 0,5Z_{t-1} + a_t$$

Setelah diperoleh 100 data dengan model yang valid dan sesuai dengan model bangkitan maka setiap data dibagi menjadi data *in sample* dan *out sample*. Dari 4900 observasi, data *out sample* yang digunakan sebanyak 100 data terakhir dan sisanya menjadi data *in sample*. Data *in sample* inilah yang kemudian akan disisipkan empat jenis *outlier* yang berbeda-beda.

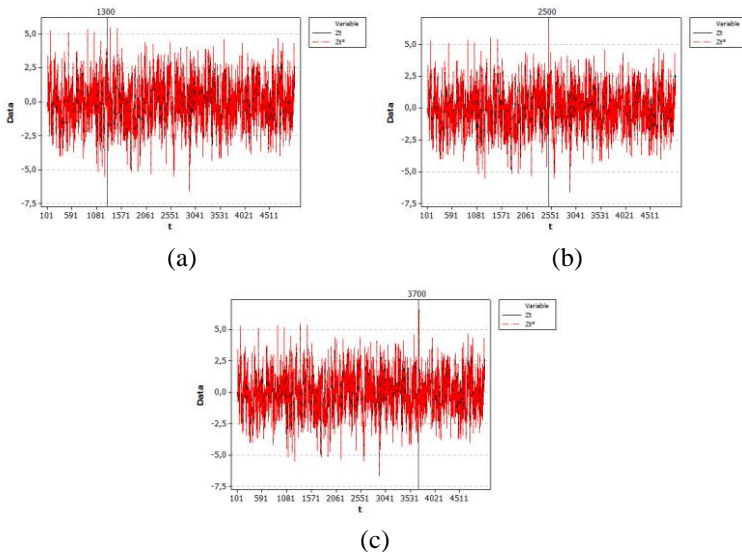
4.2 Penyisipan Outlier

Dengan menggunakan data simulasi yang sama, masing-masing disisipkan *outlier* tunggal dengan jenis yang berbeda yaitu *Additional Outlier* (AO), *Innovational Outlier* (IO), *Temporary Change* (TC) atau *Level Shift* (LS) di lokasi yang berbeda yaitu

depan ($T=1200$), tengah ($T=2400$) atau belakang ($T=3600$) dari keseluruhan data observasi.

4.2.1 Pengaruh *Additional Outlier* (AO)

Additional Outlier (AO) merupakan kejadian yang mempunyai efek pada data *time series* hanya pada satu periode tertentu saja. Pada data simulasi ini, diberikan efek *outlier* tunggal yang lokasinya di depan yaitu pada $T = 1200$, di tengah yaitu pada $T = 2400$ dan di belakang yaitu pada $T = 3600$. Besarnya efek *outlier* yang diberikan adalah $\omega = 4$. Ilustrasi *time series plot* setelah penambahan efek AO adalah sebagai berikut.

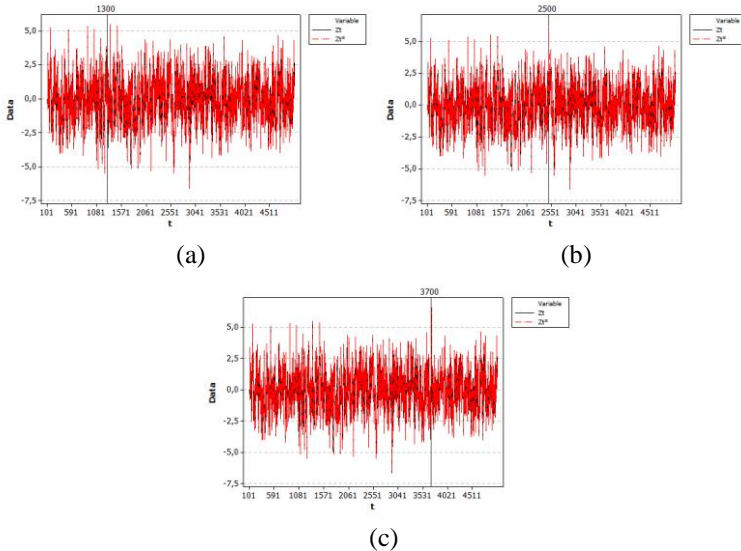


Gambar 4.2 Time Series Plot Model $Z_t = -0,8Z_{t-1} + a_t$ Perulangan Pertama Setelah Penambahan AO (a) $T = 1200$ (b) $T = 2400$ (c) $T = 3600$

4.2.2 Pengaruh *Innovational Outlier* (IO)

Innovational Outlier (IO) merupakan kejadian yang mempunyai efek sesuai dengan proses ARMA yang terdapat dalam model. Pada data simulasi ini, diberikan efek *outlier* tunggal yang lokasinya di depan yaitu pada $T = 1200$, di tengah yaitu pada

$T = 2400$ dan di belakang yaitu pada $T = 3600$. Besarnya efek *outlier* yang diberikan sesuai dengan besarnya parameter AR yang merepresentasikan setiap data. Kemudian, efek berkurang sebesar pangkat $k + 1$ seiring dengan bertambahnya 1 periode observasi. Ilustrasi *time series plot* setelah penambahan efek IO adalah sebagai berikut.

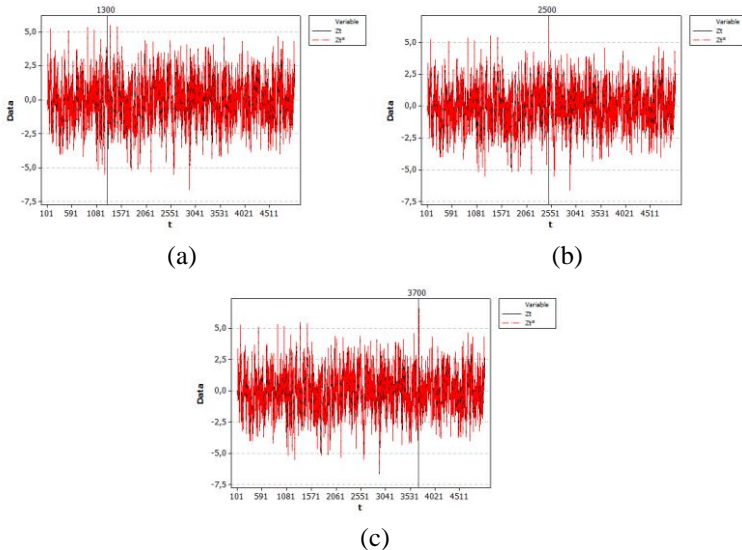


Gambar 4.3 *Time Series Plot* Model $Z_t = -0,8Z_{t-1} + a_t$ Perulangan Pertama Setelah Penambahan IO (a) $T = 1200$ (b) $T = 2400$ (c) $T = 3600$

4.2.3 Pengaruh *Temporary Change* (TC)

Temporary Change (TC) merupakan kejadian yang mempunyai efek awal sebesar ω pada suatu periode tertentu kemudian secara perlahan menurun sesuai dengan besarnya δ . Pada data simulasi ini, diberikan efek *outlier* tunggal yang lokasinya di depan yaitu pada $T = 1200$, di tengah yaitu pada $T = 2400$ dan di belakang yaitu pada $T = 3600$. Besarnya efek *outlier* yang diberikan sesuai dengan $\omega = 4$ dan $\delta = 0.7$. Kemudian, efek berkurang sebesar pangkat $k + 1$ seiring dengan

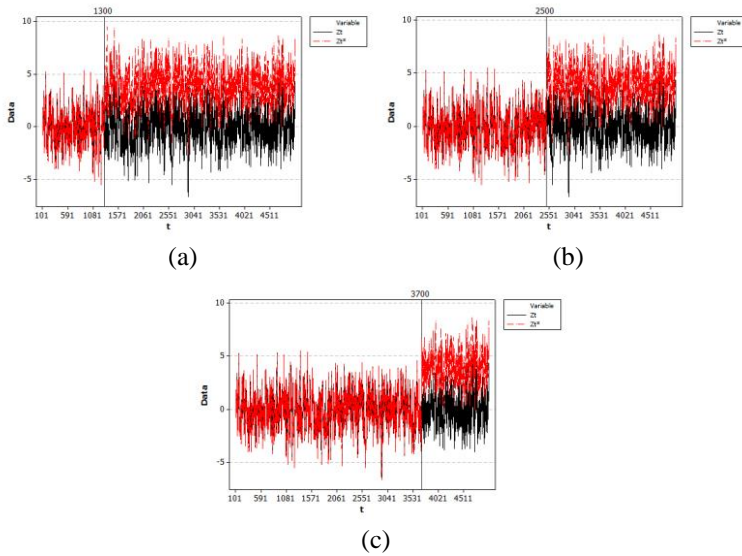
bertambahnya 1 periode observasi. Ilustrasi *time series plot* setelah penambahan efek TC adalah sebagai berikut.



Gambar 4.4 *Time Series Plot* Model $Z_t = -0,8Z_{t-1} + a_t$ Perulangan Pertama Setelah Penambahan TC (a) $T = 1200$ (b) $T = 2400$ (c) $T = 3600$

4.2.4 Pengaruh *Level Shift* (LS)

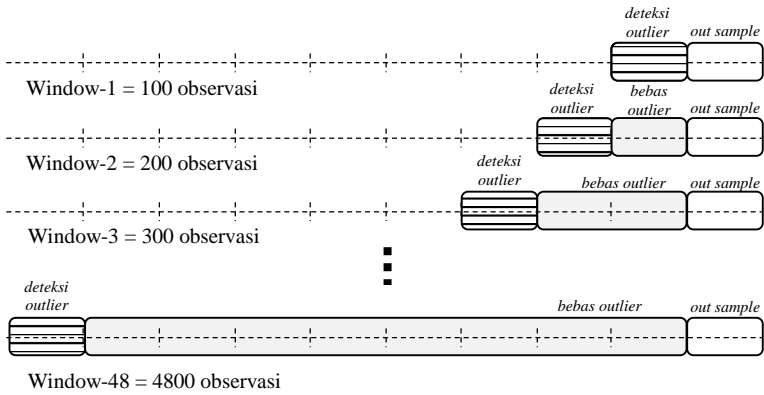
Level Shift (LS) merupakan kejadian yang mempengaruhi data *time series* pada satu waktu tertentu yang memberikan suatu perubahan tiba-tiba dan permanen. Pada data simulasi ini, diberikan efek *outlier* tunggal yang lokasinya di depan yaitu pada $T = 1200$, di tengah yaitu pada $T = 2400$ dan di belakang yaitu pada $T = 3600$. Besarnya efek *outlier* yang diberikan sesuai dengan $\omega = 4$. Ilustrasi *time series plot* setelah penambahan efek LS adalah sebagai berikut.



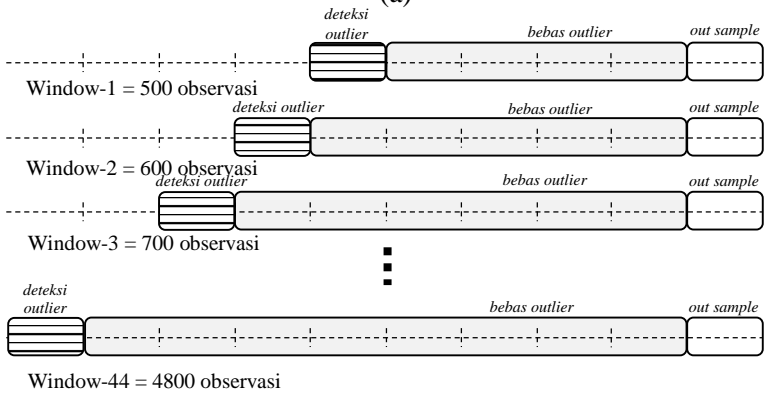
Gambar 4.5 *Time Series Plot Model $Z_t = -0,8Z_{t-1} + a_t$* Perulangan Pertama Setelah Penambahan LS (a) $T = 1200$ (b) $T = 2400$ (c) $T = 3600$

4.3 Prosedur Deteksi Outlier Dengan Metode Window Time

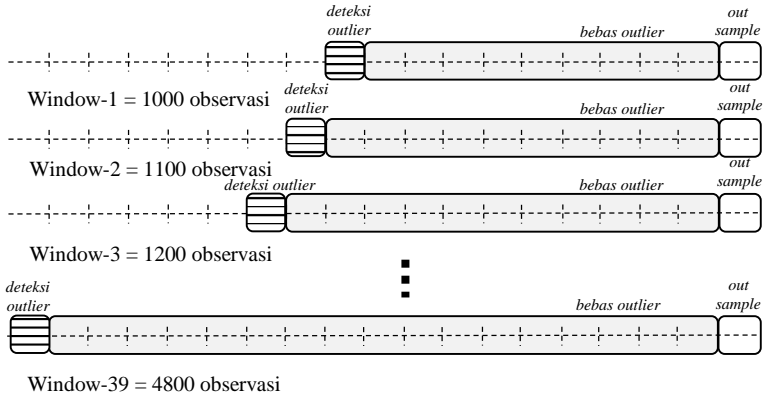
Pemodelan menggunakan *window time* pada penelitian ini memodelkan dengan menggunakan data *in sample* sesuai dengan panjang *window* yang telah ditetapkan. Pertama-tama, data yang sudah disisipi *outlier* dibagi menjadi beberapa bagian. Pada penelitian ini ditetapkan lebar pergeseran *window* sebesar 100 observasi pada setiap iterasinya. Lebar *window* awal ditetapkan sebesar 100, 500 dan 1000 observasi, sehingga terdapat 48 *window* yang dihasilkan ketika lebar *window* awal sebesar 100 observasi, 44 *window* dihasilkan ketika lebar *window* awal sebesar 500 dan 39 *window* dihasilkan ketika lebar *window* awal sebesar 1000. Berikut merupakan ilustrasi secara lengkap mengenai pergeseran *window*.



(a)



(b)



(c)

Gambar 4.6 Ilustrasi Pembagian *Window Time* dengan Lebar Window Awal (a) 100 observasi (b) 500 observasi (c) 1000 observasi

Untuk dapat memprediksi nilai data yang akan terjadi di kemudian hari, *window time* yang lokasinya dekat dengan nilai yang akan diprediksi merupakan lokasi yang paling baik untuk dapat menghasilkan akurasi yang tinggi (Hadi, 2016). Sehingga pada penelitian ini data disimulasikan untuk mendeteksi *outlier* dengan melakukan *screening* yang dimulai dari data paling baru atau data observasi yang letaknya di bagian belakang dari keseluruhan deret waktu observasi.

4.3.1 Prosedur Deteksi *Outlier* Dengan *Window Time* Awal 100

Seperti yang telah diilustrasikan sebelumnya bahwa terdapat 48 kali iterasi yang terjadi pada prosedur deteksi *outlier* dengan lebar *window* awal bebas *outlier* sebesar 100. Pada pembahasan akan dijabarkan mengenai prosedur deteksi *outlier* untuk tipe *Additional Outlier* (AO) pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Persamaan model dengan parameter tersebut dapat dituliskan sebagai $Z_t = -0,8Z_{t-1} + a_t$. Prosedur deteksi *outlier* ini dilakukan dengan tahapan sebagai berikut:

1. Jumlah data *in sample* yang digunakan dalam penelitian sebanyak 4800 observasi. Prosedur ini dilakukan pada data

simulasi yang telah disisipkan *outlier* di dalamnya baik itu AO, IO, TC maupun LS, serta menggunakan berbagai kombinasi lokasi penyisipan *outlier* di depan, tengah dan belakang. Pertama-tama, dilakukan pencarian model ARIMA terbaik dari keseluruhan data observasi.

2. Kemudian, 100 observasi terakhir dipotong untuk dideteksi *outlier* di dalamnya menggunakan prosedur deteksi *outlier* Chen dan Liu (1993).
3. Mengadaptasi dari prosedur yang dilakukan oleh Chen dan Liu (1993) di persamaan (25) dan (26), apabila terdapat *outlier* di dalam *window* tersebut maka *outlier* dihilangkan dari lebar *window time* dan kemudian dicari model ARIMA terbaik dari *window time* yang sudah bebas dari adanya *outlier*. Namun, apabila tidak ditemukan *outlier* pada *window time* awal maka *window time* tersebut langsung dicari model ARIMA terbaik untuk dapat dilakukan analisis deteksi *outlier* pada pergeseran *window* setelahnya.
4. Selanjutnya, bergeser ke *window time* berikutnya dengan lebar pergeseran sebesar 100 observasi. Sehingga lebar *window* menjadi 200 observasi. Kemudian dalam *window time* tersebut dilakukan deteksi *outlier*, apabila terdapat *outlier* di dalam *window* tersebut maka *outlier* dihilangkan dari lebar *window time* dan kemudian dicari model ARIMA terbaik dari *window time* yang sudah bebas dari adanya *window*. Namun, apabila tidak ditemukan *outlier* pada *window time* maka *window time* tersebut langsung dicari model ARIMA terbaik untuk dapat dilakukan analisis deteksi *outlier* pada pergeseran *window* setelahnya.
5. Dengan melakukan pergeseran *window time* yang sama, semua deret observasi dilakukan analisis mengenai pendeteksian *outlier* sampai iterasi ke-48.
6. Menghitung jumlah kesalahan deteksi *outlier* pada keseluruhan deret data dan menghitung prosentasenya.

Dari prosedur yang telah dijelaskan, maka akan deskripsikan mengenai salah satu hasil deteksi *outlier* yang diujicobakan dengan menyisipkan jenis *outlier* yang berbeda.

1. **Additional Outlier (AO)**

Pada penyisipan tipe *outlier* AO yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa terdapat kesalahan deteksi *outlier* pada saat data observasi ke 1201 dan 2117. Data tersebut seharusnya bukan merupakan *outlier*, namun karena kesalahan deteksi maka data pada observasi tersebut dianggap sebagai *outlier*. Sedangkan data observasi ke-1200 dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 2 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier* adalah $\frac{2}{4800} = 0,042\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan rata-rata prosentase kesalahan deteksi *outlier* sebesar 0,075% pada model dengan parameter $\phi = 0,8$

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata persentase kesalahan deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,075%, sedangkan pada parameter $\phi = 0,5$ adalah 0,049% dan pada parameter $\phi = -0,5$ adalah 0,068%.

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *window* saat $T=2400$ dan $T=3600$ pada keempat parameter model ARIMA (1,0,0) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.2.

Tabel 4.2 Prosentase Kesalahan Deteksi *Outlier* AO dengan Panjang *Initial Window* 100

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,075%
	-0,8	0,075%
	0,5	0,049%
	-0,5	0,068%
T=2400	0,8	0,036%
	-0,8	0,052%
	0,5	0,026%
	-0,5	0,051%
T=3600	0,8	0,060%
	-0,8	0,058%
	0,5	0,025%
	-0,5	0,049%

2. **Innovational *Outlier* (IO)**

Pada penyisipan tipe *outlier* IO yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa terdapat kesalahan deteksi *outlier* pada saat data observasi ke 2117. Data tersebut seharusnya bukan merupakan *outlier*, namun karena kesalahan deteksi maka data pada observasi tersebut dianggap sebagai *outlier*. Sedangkan data observasi ke-1200 tidak dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 2 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier* adalah $\frac{2}{4800} = 0,042\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan rata-rata prosentase kesalahan deteksi *outlier* sebesar 0,040% pada model dengan parameter $\phi = 0,8$

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata prosentase kesalahan

deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,038%, sedangkan pada parameter $\phi = 0,5$ adalah 0,035% dan pada parameter $\phi = -0,5$ adalah 0,051%,

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *outlier* saat $T=2400$ dan $T=3600$ pada keempat parameter model AR(1) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.3.

Tabel 4.3 Prosentase Kesalahan Deteksi *Outlier* IO dengan Panjang *Initial Window* 100

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,040%
	-0,8	0,038%
	0,5	0,035%
	-0,5	0,051%
T=2400	0,8	0,034%
	-0,8	0,034%
	0,5	0,031%
	-0,5	0,021%
T=3600	0,8	0,035%
	-0,8	0,028%
	0,5	0,027%
	-0,5	0,045%

3. Temporary Change (TC)

Pada penyisipan tipe *outlier* TC yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa tidak terdeteksi *outlier* sama sekali di dalam deret data padahal disisipkan jenis *outlier* TC di dalamnya. Data observasi ke-1200 tidak dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 1 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier* adalah $\frac{1}{4800} = 0,021\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan rata-rata

prosentase kesalahan deteksi *outlier* sebesar 0,034% pada model dengan parameter $\phi = 0,8$

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata persentase kesalahan deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,034%, sedangkan pada parameter $\phi = 0,5$ adalah 0,032% dan pada parameter $\phi = -0,5$ adalah 0,048%

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *outlier* saat $T=2400$ dan $T=3600$ pada keempat parameter model AR(1) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.4.

Tabel 4.4 Prosentase Kesalahan Deteksi *Outlier* TC dengan Panjang *Initial Window* 100

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,034%
	-0,8	0,034%
	0,5	0,032%
	-0,5	0,048%
T=2400	0,8	0,041%
	-0,8	0,029%
	0,5	0,028%
	-0,5	0,045%
T=3600	0,8	0,023%
	-0,8	0,021%
	0,5	0,021%
	-0,5	0,021%

4.3.2 Prosedur Deteksi *Outlier* Dengan *Window Time* Awal 500

Terdapat 44 kali iterasi yang terjadi pada prosedur deteksi *outlier* dengan lebar *window* awal bebas *outlier* sebesar 500. Pada pembahasan akan dijabarkan mengenai prosedur deteksi *outlier* untuk tipe *Additional Outlier* (AO) pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Persamaan model dengan parameter tersebut dapat

dituliskan sebagai $Z_t = -0,8Z_{t-1} + a_t$. Prosedur deteksi *outlier* ini dilakukan dengan tahapan sebagai berikut:

1. Jumlah data in sample yang digunakan dalam penelitian sebanyak 4800 observasi. Prosedur ini dilakukan pada data simulasi yang telah disisipkan *outlier* di dalamnya baik itu AO, IO, TC maupun LS, serta menggunakan berbagai kombinasi lokasi penyisipan *outlier* di depan, tengah dan belakang. Pertama-tama, dilakukan pencarian model ARIMA terbaik dari keseluruhan data observasi.
2. Kemudian, 500 observasi terakhir dipotong untuk dideteksi *outlier* di dalamnya menggunakan prosedur deteksi *outlier* Chen dan Liu (1993).
3. Mengadaptasi dari prosedur yang dilakukan oleh Chen dan Liu (1993) di persamaan (25) dan (26), apabila terdapat *outlier* di dalam *window* tersebut maka *outlier* dihilangkan dari lebar *window time* dan kemudian dicari model ARIMA terbaik dari *window time* yang sudah bebas dari adanya *outlier*. Namun, apabila tidak ditemukan *outlier* pada *window time* awal maka *window time* tersebut langsung dicari model ARIMA terbaik untuk dapat dilakuka analisis deteksi *outlier* pada pergeseran *window* setelahnya.
4. Selanjutnya, bergeser ke *window time* berikutnya dengan lebar pergeseran sebesar 100 observasi. Sehingga lebar *window* menjadi 200 observasi. Kemudian dalam *window time* tersebut dilakukan deteksi *outlier*, apabila terdapat *outlier* di dalam *window* tersebut maka *outlier* dihilangkan dari lebar *window time* dan kemudian dicari model ARIMA terbaik dari *window time* yang sudah bebas dari adanya *outlier*. Namun, apabila tidak ditemukan *outlier* pada *window time* maka *window time* tersebut langsung dicari model ARIMA terbaik untuk dapat dilakukan analisis deteksi *outlier* pada pergeseran *window* setelahnya.
5. Dengan melakukan pergeseran *window time* yang sama, semua deret observasi dilakukan analisis mengenai pendeteksian *outlier* sampai iterasi ke-44.

6. Menghitung jumlah kesalahan deteksi *outlier* pada keseluruhan deret data dan menghitung prosentasenya.

Dari prosedur yang telah dijelaskan, maka akan deskripsikan mengenai salah satu hasil deteksi *outlier* yang diujicobakan dengan menyisipkan jenis *outlier* yang berbeda.

1. **Additional Outlier (AO)**

Pada penyisipan tipe *outlier* AO yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa terdapat kesalahan deteksi *outlier* pada saat data observasi ke 1201 dan 2117. Data tersebut seharusnya bukan merupakan *outlier*, namun karena kesalahan deteksi maka data pada observasi tersebut dianggap sebagai *outlier*. Sedangkan data observasi ke-1200 dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 2 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier* adalah $\frac{2}{4800} = 0,042\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan prosentase kesalahan deteksi *outlier* sebesar 0,082% pada model dengan parameter $\phi = 0,8$.

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata persentase kesalahan deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,073%, sedangkan pada parameter $\phi = 0,5$ adalah 0,048% dan pada parameter $\phi = -0,5$ adalah 0,059%,

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *outlier* saat $T=2400$ dan $T=3600$ pada keempat parameter model AR(1) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.5.

Tabel 4.5 Prosentase Kesalahan Deteksi *Outlier* AO dengan Panjang *Initial Window* 500

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,082%
	-0,8	0,073%
	0,5	0,048%
	-0,5	0,059%
T=2400	0,8	0,035%
	-0,8	0,051%
	0,5	0,025%
	-0,5	0,050%
T=3600	0,8	0,082%
	-0,8	0,058%
	0,5	0,023%
	-0,5	0,047%

2. **Innovational *Outlier* (IO)**

Pada penyisipan tipe *outlier* IO yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa terdapat kesalahan deteksi *outlier* pada saat data observasi ke 2117. Data tersebut seharusnya bukan merupakan *outlier*, namun karena kesalahan deteksi maka data pada observasi tersebut dianggap sebagai *outlier*. Sedangkan data observasi ke-1200 tidak dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 2 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier* adalah $\frac{2}{4800} = 0,042\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan rata-rata prosentase kesalahan deteksi *outlier* sebesar 0,022% pada model dengan parameter $\phi = 0,8$

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata persentase kesalahan

deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,038%, sedangkan pada parameter $\phi = 0,5$ adalah 0,024% dan pada parameter $\phi = -0,5$ adalah 0,021%,

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *outlier* saat $T=2400$ dan $T=3600$ pada keempat parameter model AR(1) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.6.

Tabel 4.6 Prosentase Kesalahan Deteksi *Outlier* IO dengan Panjang Initial Window 500

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,022%
	-0,8	0,038%
	0,5	0,024%
	-0,5	0,021%
T=2400	0,8	0,033%
	-0,8	0,033%
	0,5	0,030%
	-0,5	0,021%
T=3600	0,8	0,034%
	-0,8	0,027%
	0,5	0,026%
	-0,5	0,029%

3. Temporary Change (TC)

Pada penyisipan tipe *outlier* TC yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa tidak terdeteksi *outlier* sama sekali di dalam deret data padahal disisipkan jenis *outlier* TC di dalamnya. Data observasi ke-1200 tidak dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 1 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier* adalah $\frac{1}{4800} = 0,021\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan rata-rata

prosentase kesalahan deteksi *outlier* sebesar 0,034% pada model dengan parameter $\phi = 0,8$

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata persentase kesalahan deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,034%, sedangkan pada parameter $\phi = 0,5$ adalah 0,031% dan pada parameter $\phi = -0,5$ adalah 0,021%

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *outlier* saat $T=2400$ dan $T=3600$ pada keempat parameter model AR(1) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.7.

Tabel 4.7 Prosentase Kesalahan Deteksi *Outlier* TC dengan Panjang *Initial Window* 500

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,034%
	-0,8	0,034%
	0,5	0,031%
	-0,5	0,021%
T=2400	0,8	0,041%
	-0,8	0,029%
	0,5	0,028%
	-0,5	0,045%
T=3600	0,8	0,021%
	-0,8	0,021%
	0,5	0,021%
	-0,5	0,021%

4.3.3 Prosedur Deteksi *Outlier* Dengan *Window Time* Awal 1000

Terdapat 39 kali iterasi yang terjadi pada prosedur deteksi *outlier* dengan lebar *window* awal bebas *outlier* sebesar 1000. Pada pembahasan akan dijabarkan mengenai prosedur deteksi *window* untuk tipe Additional *Outlier* (AO) pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan

pertama. Persamaan model dengan parameter tersebut dapat dituliskan sebagai $Z_t = -0,8Z_{t-1} + a_t$. Prosedur deteksi *outlier* ini dilakukan dengan tahapan sebagai berikut:

1. Jumlah data in sample yang digunakan dalam penelitian sebanyak 4800 observasi. Prosedur ini dilakukan pada data simulasi yang telah disisipkan *outlier* di dalamnya baik itu AO, IO, TC maupun LS, serta menggunakan berbagai kombinasi lokasi penyisipan *outlier* di depan, tengah dan belakang. Pertama-tama, dilakukan pencarian model ARIMA terbaik dari keseluruhan data observasi.
2. Kemudian, 1000 observasi terakhir dipotong untuk dideteksi *outlier* di dalamnya menggunakan prosedur deteksi *outlier* Chen dan Liu (1993).
3. Mengadaptasi dari prosedur yang dilakukan oleh Chen dan Liu (1993) di persamaan (25) dan (26), apabila terdapat *outlier* di dalam *window* tersebut maka *outlier* dihilangkan dari lebar *window time* dan kemudian dicari model ARIMA terbaik dari *window time* yang sudah bebas dari adanya *outlier*. Namun, apabila tidak ditemukan *outlier* pada *window time* awal maka *window time* tersebut langsung dicari model ARIMA terbaik untuk dapat dilakukan analisis deteksi *outlier* pada pergeseran *window* setelahnya.
4. Selanjutnya, bergeser ke *window time* berikutnya dengan lebar pergeseran sebesar 100 observasi. Sehingga lebar *window* menjadi 200 observasi. Kemudian dalam *window time* tersebut dilakukan deteksi *outlier*, apabila terdapat *outlier* di dalam *window* tersebut maka *outlier* dihilangkan dari lebar *window time* dan kemudian dicari model ARIMA terbaik dari *window time* yang sudah bebas dari adanya *outlier*. Namun, apabila tidak ditemukan *outlier* pada *window time* maka *window time* tersebut langsung dicari model ARIMA terbaik untuk dapat dilakukan analisis deteksi *outlier* pada pergeseran *window* setelahnya.

5. Dengan melakukan pergeseran *window time* yang sama, semua deret observasi dilakukan analisis mengenai pendeteksian *outlier* sampai iterasi ke-39.
6. Menghitung jumlah kesalahan deteksi *outlier* pada keseluruhan deret data dan menghitung prosentasenya.

Dari prosedur yang telah dijelaskan, maka akan deskripsikan mengenai salah satu hasil deteksi *outlier* yang diujicobakan dengan menyisipkan jenis *outlier* yang berbeda.

1. **Additional Outlier (AO)**

Pada penyisipan tipe *outlier* AO yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa terdapat kesalahan deteksi *outlier* pada saat data observasi ke 1201 dan 2117. Data tersebut seharusnya bukan merupakan *outlier*, namun karena kesalahan deteksi maka data pada observasi tersebut dianggap sebagai *outlier*. Sedangkan data observasi ke-1200 dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 2 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier* adalah $\frac{2}{4800} = 0,042\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan prosentase kesalahan deteksi *window* sebesar 0,081% pada model dengan parameter $\phi = 0,8$.

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata persentase kesalahan deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,074%, sedangkan pada parameter $\phi = 0,5$ adalah 0,048% dan pada parameter $\phi = -0,5$ adalah 0,064%.

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *outlier* saat $T=2400$ dan $T=3600$ pada

keempat parameter model AR(1) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.8.

Tabel 4.8 Prosentase Kesalahan Deteksi *Outlier* AO dengan Panjang *Initial Window* 500

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,081%
	-0,8	0,074%
	0,5	0,048%
	-0,5	0,064%
T=2400	0,8	0,035%
	-0,8	0,050%
	0,5	0,025%
	-0,5	0,049%
T=3600	0,8	0,058%
	-0,8	0,056%
	0,5	0,023%
	-0,5	0,046%

2. *Innovational Outlier (IO)*

Pada penyisipan tipe *outlier* IO yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa terdapat kesalahan deteksi *outlier* pada saat data observasi ke 2117. Data tersebut seharusnya bukan merupakan *outlier*, namun karena kesalahan deteksi maka data pada observasi tersebut dianggap sebagai *outlier*. Sedangkan data observasi ke-1200 tidak dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 2 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier* adalah $\frac{2}{4800} = 0,042\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan rata-rata prosentase kesalahan deteksi *outlier* sebesar 0,036% pada model dengan parameter $\phi = 0,8$

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata persentase kesalahan deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,037%, sedangkan pada parameter $\phi = 0,5$ adalah 0,033% dan pada parameter $\phi = -0,5$ adalah 0,021%.

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *outlier* saat $T=2400$ dan $T=3600$ pada keempat parameter model AR(1) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.9.

Tabel 4.9 Prosentase Kesalahan Deteksi *Ooutlier* IO dengan Panjang *Initial Window* 500

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,036%
	-0,8	0,037%
	0,5	0,033%
	-0,5	0,021%
T=2400	0,8	0,033%
	-0,8	0,032%
	0,5	0,029%
	-0,5	0,021%
T=3600	0,8	0,028%
	-0,8	0,026%
	0,5	0,027%
	-0,5	0,027%

3. Temporary Change (TC)

Pada penyisipan tipe *outlier* TC yang diletakkan pada data observasi sebesar $\omega = 4$ pada saat observasi ke 1200 pada model ARIMA (1,0,0) dengan parameter $\phi = 0,8$ pada model bangkitan perulangan pertama. Didapatkan hasil bahwa tidak terdeteksi *outlier* sama sekali di dalam deret data padahal disisipkan jenis *outlier* TC di dalamnya. Data observasi ke-1200 tidak dideteksi secara benar sebagai *outlier*. Dari panjang data $n=4800$ terdapat 1 kesalahan deteksi *outlier* sehingga prosentase kesalahan deteksi *outlier*

adalah $\frac{1}{4800} = 0,021\%$. Selanjutnya dilakukan prosedur yang sama untuk model perulangan berikutnya sampai pada data perulangan ke 100. Prosedur ini menghasilkan rata-rata prosentase kesalahan deteksi *outlier* sebesar 0,034% pada model dengan parameter $\phi = 0,8$

Selanjutnya dilakukan prosedur yang sama untuk parameter model AR(1) yang berbeda yaitu $\phi = -0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Hasil rata-rata persentase kesalahan deteksi *outlier* pada parameter $\phi = -0,8$ adalah 0,033%, sedangkan pada parameter $\phi = 0,5$ adalah 0,031% dan pada parameter $\phi = -0,5$ adalah 0,021%

Selain itu deteksi *outlier* juga dilakukan pada data yang disisipkan *outlier* saat $T=2400$ dan $T=3600$ pada keempat parameter model AR(1) yang berbeda. Hasil prosentase kesalahan deteksi disajikan dalam Tabel 4.10.

Tabel 4.10 Prosentase Kesalahan Deteksi *Outlier* TC dengan Panjang *Initial Window* 500

Lokasi <i>Outlier</i>	Parameter (ϕ)	Kesalahan Deteksi (%)
T=1200	0,8	0,034%
	-0,8	0,033%
	0,5	0,031%
	-0,5	0,021%
T=2400	0,8	0,021%
	-0,8	0,021%
	0,5	0,021%
	-0,5	0,021%
T=3600	0,8	0,021%
	-0,8	0,022%
	0,5	0,022%
	-0,5	0,021%

4.3.4 Pembahasan *Outlier* Jenis *Level Shift*

Jenis *outlier level shift* merupakan kejadian yang mempengaruhi deret pada suatu waktu tertentu dan efek dari *outlier* tersebut membuat suatu perubahan yang tiba-tiba dan

permanen sampai akhir periode. Metode yang paling baik untuk mengatasi jenis *outlier* ini adalah dengan menggunakan analisis intervensi *step function* karena dapat memodelkan pola data yang besarnya berubah secara permanen. Sedangkan dalam penelitian ini cara yang digunakan untuk mengatasi ketiga jenis *outlier* yang lain adalah dengan menghilangkan data yang terdeteksi sebagai *outlier* (Chen dan Liu, 1993). Sehingga untuk analisis deteksi *outlier* pada prosedur *window time* yang ada dalam penelitian ini tidak membahas hasil data simulasi yang disisipkan dengan *outlier* jenis level shift.

4.3.5 Pengujian *Multi Way* ANOVA untuk Mengetahui Pengaruh Lebar *Window Time* Awal Terhadap Persentase Kesalahan Deteksi *Outlier*

Dalam penelitian ini terdapat 4 faktor yang diduga berpengaruh terhadap kesalahan deteksi *outlier* yang terdapat pada data simulasi. Faktor pertama adalah parameter model ARIMA (1,0,0) yang dibangkitkan yaitu 0,8; -0,8; 0,5 dan -0,5. Faktor kedua adalah jenis *outlier* yang terdapat pada data yaitu AO, IO dan TC. Faktor ketiga adalah panjang *window time* awal yang dideteksi keberadaan *window*nya yaitu 100, 500 dan 1000. Dan faktor yang terakhir adalah lokasi keberadaan *outlier* yang disisipkan yaitu berada di depan (T=1200), tengah (T=2400) dan belakang (T=3600). Untuk menguji apakah keempat faktor yang disebutkan diatas berpengaruh terhadap kesalahan deteksi *outlier* dilakukan pengujian *Multi Way* ANOVA terhadap hasil data kesalahan deteksi *outlier*.

1. Rata-Rata Kesalahan Deteksi *Outlier* Berdasarkan Lebar *Window Time* Awal

Salah satu faktor yang menjadi objek penelitian adalah pengaruh panjang awal *window time* terhadap kesalahan deteksi *outlier*. Tabel 4.11 merupakan rata-rata kesalahan deteksi *outlier* berdasarkan panjang *window time* awal yang diujikan yaitu 100, 500 dan 1000.

Tabel 4.11 Rata-Rata Kesalahan Deteksi *Outlier* Berdasarkan Lebar *Window Time* Awal

No.	<i>Window Time</i> Awal	Rata-Rata (%)
1.	100	0,03957
2.	500	0,03445
3.	1000	0,03473

Prosentase rata-rata kesalahan deteksi *outlier* yang terjadi ketika dicobakan dengan lebar *window time* awal 100 adalah 0,03957%, selanjutnya menurun ketika dicobakan pada *window time* yang lebih lebar yaitu 500 dengan rata-rata prosentase kesalahan deteksi sebesar 0,03445%. Ketika lebar *window time* sebesar 1000 menghasilkan prosentase sebesar 0,03473%.

2. Uji Homogenitas

Salah satu asumsi yang diperlukan dalam pengujian *Multi Way* ANOVA adalah varians antar kelompok harus bersifat homogen. Untuk menguji kehomogenan varians antar kelompok digunakan *Levene's Test* seperti ditunjukkan pada Tabel 4.12 berikut.

Tabel 4.12 *Levene's Test* Untuk Menguji Homogenitas

F	df1	df2	Sig.
13,622	107	10692	0,000

Tabel 4.10 diatas menunjukkan bahwa nilai signifikansi sebesar 0,000 yaitu kurang dari nilai $\alpha = 0,05$, sehingga dapat dikatakan varians antar kelompok secara signifikan bersifat homogen. Sehingga dapat dilakukan uji *Multi Way* ANOVA.

3. Multi Way ANOVA

Pengujian *Multi Way* ANOVA dilakukan untuk mengetahui faktor-faktor apa saja yang mempengaruhi kesalahan deteksi *outlier* yang dilakukan pada data simulasi. Dalam penelitian ini diduga terdapat 4 faktor yang mempengaruhi kesalahan deteksi *outlier* yaitu besarnya parameter dalam model, jenis *outlier* yang ada dalam deret, lebar *window time* awal dan lokasi keberadaan *outlier*. Tabel 4.13 berikut menunjukkan hasil pengujian *multi way* ANOVA.

Tabel 4.13 Hasil *Multi Way* ANOVA

<i>Source</i>	<i>Type III Sum of Squares</i>	<i>Sig</i>
<i>Corrected Model</i>	2,623	0,000
<i>Intercept</i>	14,191	0,000
Paramater	0,207	0,000
Jenis <i>Outlier</i>	1,263	0,000
Lebar <i>Window Time</i> Awal	0,060	0,000
Lokasi <i>Outlier</i>	0,323	0,000
Paramater * Jenis <i>Outlier</i>	0,251	0,000
Paramater * Lebar <i>Window Time</i> Awal	0,041	0,018
Paramater * Jenis <i>Outlier</i>	0,029	0,095
Jenis <i>Outlier</i> * Lebar <i>Window Time</i> Awal	0,018	0,143
Jenis <i>Outlier</i> * Lokasi <i>Outlier</i>	0,196	0,000
Lebar <i>Window Time</i> Awal * Lokasi <i>Outlier</i>	0,010	0,443
Paramater * Jenis <i>Outlier</i> * Lebar <i>Window Time</i> Awal	0,015	0,932
Paramater * Jenis <i>Outlier</i> * Lokasi <i>Outlier</i>	0,116	0,000
Paramater * Lebar <i>Window Time</i> Awal * Lokasi <i>Outlier</i>	0,025	0,678
Jenis <i>Outlier</i> * Lebar <i>Window Time</i> Awal * Lokasi <i>Outlier</i>	0,044	0,034
Paramater * Jenis <i>Outlier</i> * Lebar <i>Window Time</i> Awal * Lokasi <i>Outlier</i>	0,025	0,996

Berdasarkan nilai *corrected model* dapat disimpulkan bahwa semua variabel independen yaitu besarnya parameter dalam model, jenis outlier yang ada dalam deret, lebar *window time* awal dan lokasi keberadaan *outlier* secara serentak berpengaruh terhadap prosentase kesalahan deteksi *outlier*. Hal ini ditunjukkan dengan nilai signifikansi sebesar 0,000 yaitu kurang dari nilai $\alpha = 0,05$, sehingga dapat dikatakan bahwa model tersebut valid.

Nilai *intercept* menunjukkan perubahan prosentase kesalahan deteksi *outlier* yang tidak dipengaruhi keberadaan

variabel independen yaitu keempat faktor yang diduga berpengaruh seperti disebutkan diatas. Artinya, tanpa adanya pengaruh variabel independen, prosentase kesalahan deteksi *outlier* dapat berubah nilainya. Pada penelitian *intercept* dalam model berpengaruh signifikan ditunjukkan dengan nilai signifikansi sebesar 0,000 yaitu kurang dari nilai $\alpha = 0,05$.

Nilai signifikansi dari empat faktor yang diduga berpengaruh terhadap prosentase kesalahan deteksi *outlier* bernilai 0,000 yaitu kurang dari nilai $\alpha = 0,05$, sehingga berarti bahwa besarnya parameter dalam model, jenis outlier yang ada dalam deret, lebar *window time* awal dan lokasi keberadaan outlier berpengaruh signifikan terhadap kesalahan deteksi *outlier*. Parameter dalam model, jenis *outlier* dan lokasi keberadaan *outlier* merupakan faktor-faktor yang tidak bisa diubah dalam suatu data riil. Faktor-faktor tersebut menjadi suatu karakteristik masing-masing yang menjadi ciri khas sebuah data. Dalam penelitian ini akan dibandingkan mengenai faktor lebar *window time* awal yang dapat diubah-ubah sesuai dengan penelitian.

Interaksi antar faktor yang berpengaruh signifikan terhadap kesalahan deteksi *outlier* adalah parameter * lebar *window time* awal dengan nilai signifikansi sebesar 0,018, jenis *outlier* * lokasi *outlier* dengan nilai signifikansi sebesar 0,000, parameter * jenis *outlier* * lokasi *outlier* dengan nilai signifikansi sebesar 0,000 dan jenis *outlier* * lebar *window time* awal * lokasi *outlier* dengan nilai signifikansi sebesar 0,034. Sedangkan interaksi lainnya tidak berpengaruh signifikan terhadap kesalahan deteksi outlier. Sebagai contoh, interaksi yang mengandung lokasi *outlier* dan lebar *window time* awal cenderung tidak signifikan karena pada pengamatan outlier diletakkan di luar 1000 observasi terakhir sedangkan lebar *window time* paling maksimum adalah 1000 observasi terakhir. Secara ideal, hal ini membuktikan bahwa pada semua lebar *window time* awal tidak akan dideteksi outlier sehingga tidak berpengaruh signifikan terhadap kesalahan deteksi outlier.

4. *Tukey Post Hoc*

Dengan menggunakan Uji *Tukey* dapat diketahui kategori manakah dari lebar *window time* awal yang memiliki perbedaan secara signifikan. Tabel 4.14 berikut menunjukkan hasil dari Uji *Tukey*.

Tabel 4.14 Hasil *Uji Tukey Post Hoc*

Lebar <i>window time</i> awal	Lebar <i>window time</i> awal	Selisih Rata-Rata	Sig.
100	500	0,00512	0,000
	1000	0,00483	0,000
500	100	-0,00512	0,000
	1000	-0,00029	0,969
1000	100	-0,00483	0,000
	500	0,00029	0,969

Dari Tabel 4.14 di atas dapat dilihat bahwa terdapat perbedaan signifikan antara lebar *window time* awal 100 dengan 500 dan 100 dengan 1000 dengan nilai signifikansi sebesar 0,000 yaitu kurang dari nilai $\alpha = 0,05$. Sehingga selanjutnya perlu diteliti tentang rata-rata akurasi masing-masing lebar *window time* awal. Tabel 4.14 menjelaskan bahwa rata-rata prosentase kesalahan deteksi *outlier* kelompok dengan lebar *window time* awal 500 sebesar 0,03445% tidak berbeda secara signifikan dengan rata-rata prosentase kesalahan deteksi *outlier* kelompok dengan lebar *window time* awal 1000 sebesar 0,03473%. Sedangkan rata-rata prosentase kesalahan deteksi *outlier* kelompok dengan lebar *window time* awal 100 yaitu sebesar 0,03957% berbeda secara signifikan dengan rata-rata prosentase kesalahan deteksi *outlier* kelompok dengan lebar *window time* awal 500 dan 1000. Karena nilai prosentase lebar *window time* awal 100 lebih besar dibandingkan dengan nilai prosentase lebar *window time* awal 500 dan 1000, maka lebar *window time* awal 500 dan 1000 memberikan akurasi yang lebih baik.

4.4 Membandingkan Akurasi Hasil Prediksi

Setelah mendapatkan hasil bahwa dengan lebar *window time* awal 500 dan 1000 memberikan nilai rata-rata prosentase kesalahan deteksi *outlier* yang lebih baik dibandingkan dengan lebar *window time* awal 100. Selanjutnya akan dibandingkan akurasi hasil prediksi dari data *out sampel* sebanyak 100 observasi yang akan digunakan dengan 3 cara yaitu: (1) prediksi tanpa melakukan deteksi *outlier* pada data, (2) prediksi dengan melakukan deteksi *outlier* di keseluruhan data, dan (3) prediksi dengan melakukan deteksi *outlier* dan *window time*. Perhitungan akurasi dari nilai prediksi menggunakan nilai RMSE. Nilai prediksi akan semakin akurat apabila nilai RMSE yang dihasilkan semakin kecil. Hasil perbandingan akurasi dari prediksi menggunakan RMSE ditunjukkan pada Tabel 4.15 berikut.

Tabel 4.15 Hasil Perbandingan Nilai RMSE

Jenis Outlier	Lokasi Outlier	Parameter	Cara Terbaik
AO	Depan	0,8	3
		-0,8	2
		0,5	3
		-0,5	3
	Tengah	0,8	3
		-0,8	1
		0,5	3
		-0,5	3
	Belakang	0,8	3
		-0,8	1
		0,5	3
		-0,5	3
IO	Depan	0,8	3
		-0,8	2
		0,5	3
		-0,5	3
	Tengah	0,8	3
		-0,8	2
		0,5	3
		-0,5	3
	Belakang	0,8	3
		-0,8	2
		0,5	3

TC	Depan	-0,5	3
		0,8	3
		-0,8	2
		0,5	3
		-0,5	3
	Tengah	0,8	3
		-0,8	2
		0,5	3
		-0,5	3
		0,8	3
	Belakang	-0,8	1
		0,5	3
		-0,5	3

Hasil perbandingan ketiga cara memberikan kesimpulan bahwa cara ketiga yaitu prediksi dengan melakukan deteksi outlier dan window time menghasilkan RMSE yang paling kecil pada model pertama, ketiga dan kedua yaitu $Z_t = -0,8Z_{t-1} + a_t$, $Z_t = -0,5Z_{t-1} + a_t$ dan $Z_t = 0,5Z_{t-1} + a_t$. Sedangkan pada model kedua yaitu $Z_t = 0,8Z_{t-1} + a_t$ dengan parameter model -0,8, cara ketiga tidak menghasilkan nilai RMSE yang paling kecil dibandingkan kedua cara yang lainnya. Sehingga dapat disimpulkan deteksi outlier dengan menggunakan window time menghasilkan akurasi yang baik jika parameter model $\phi = 0,8$, $\phi = 0,5$ dan $\phi = -0,5$. Pada penelitian ini hanya dicobakan pada keempat nilai parameter itu saja, namun tidak menutup kemungkinan untuk memberikan hasil pada parameter-parameter selain yang disebutkan untuk diteliti pada penelitian selanjutnya.

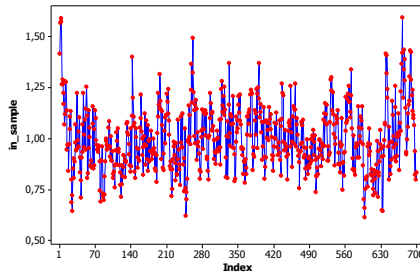
4.5 Studi Kasus (*Tree Rings*)

Untuk mengamati apakah dengan digunakan data riil yang sebenarnya akan dihasilkan kesimpulan yang sama dengan menggunakan data simulasi, maka data riil yang akan digunakan adalah data lingkaran pohon yang ada di Chili. Data ini digunakan karena diduga memiliki model ARIMA yang sama dengan data simulasi yaitu ARIMA (1,0,0). Data tersedia dalam *website* resmi www.datamarket.com dalam kategori *tree rings*. Lingkaran pohon

diukur dan dicatat setiap tahun sejak tahun 1242 sampai dengan 1975, sehingga terdapat 734 observasi. Namun karena terdapat penurunan yang sangat tinggi pada tahun 1262 dan 1263 sehingga diduga mengakibatkan berubahnya pola dan model data menjadi bukan merupakan ARIMA (1,0,0) maka data yang dijadikan observasi untuk pengujian studi kasus adalah tahun 1264 sampai dengan 1975. Dengan begitu terdapat 712 observasi yang diamati dalam *time series*. Selanjutnya 712 observasi tersebut dibagi menjadi 700 observasi *in sample* dan 12 observasi *out sample*. Pembagian ini ditentukan berdasarkan prosentase pembagian data *in sample* dan *out sample* yang dilakukan pada data simulasi, selain itu untuk memudahkan pemotongan *window time* yang dilakukan pada data observasi dengan pergeseran sebesar 100 observasi. Selanjutnya dilakukan spesifikasi model dengan tahap-tahap berikut.

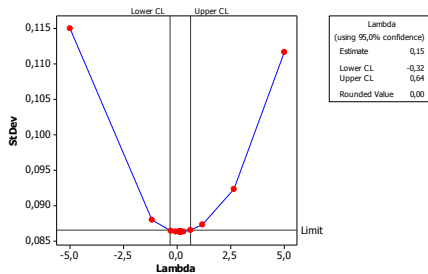
4.5.1 Identifikasi Model

Kestasioneran data *time series* dibagi menjadi dua yaitu stasioner dalam *mean* dan stasioner dalam varians. Secara visual, stasioner dalam *mean* dapat dilihat menggunakan *times series plot* dan diagram *autocorrelation function* (ACF), namun secara pengujian dapat dilakukan *Augmented Dickey-Fuller Test* untuk mendapatkan hasil yang akurat. Sedangkan stasioner dalam varians dapat dilihat dari *Box-Cox plot*. Pada *times series plot* data dikatakan stasioner dalam *mean* apabila data berfluktuasi di sekitar suatu nilai *mean* yang konstan, jika dilihat dari diagram ACF data dikatakan tidak stasioner dalam *mean* apabila lag-lag pada diagram ACF turun secara lambat. Data dikatakan stasioner dalam varians apabila nilai *rounded value* sama dengan 1 atau batas bawah dan batas atas telah melewati angka 1. Berikut merupakan *time series plot* dari data *tree rings*.



Gambar 4.7 *Time Series Plot Data Tree Rings*

Secara visual data sudah stasioner terhadap varians dan mean, namun untuk mendapatkan pengujian yang tepat maka dibuat *Box-Cox plot* untuk melihat kestasioneran data terhadap varians sebagai berikut.

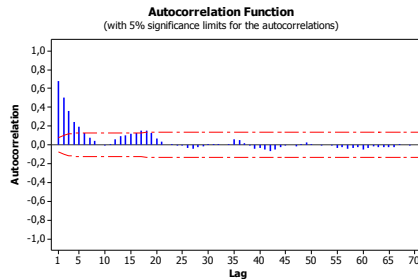


Gambar 4.8 *Box-Cox Plot Data Tree Rings*

Ternyata diperoleh hasil bahwa data tidak stasioner terhadap varians. Ditunjukkan dengan nilai rounded value yang tidak sama dengan 1 dan antara nilai batas atas dan batas bawah tidak melewati angka 1, sehingga data harus ditransformasi untuk mendapatkan data yang stasioner terhadap varians dengan transformasi baru sebagai berikut

$$Z_t^* = \ln(Z_t)$$

dimana Z_t^* adalah data setelah ditransformasi dan Z_t adalah data sebelum ditransformasi. Selanjutnya untuk mengecek kestasioneran data terhadap mean dibuat *ACF plot* dengan melihat penurunan lag-lag yang ada didalamnya.



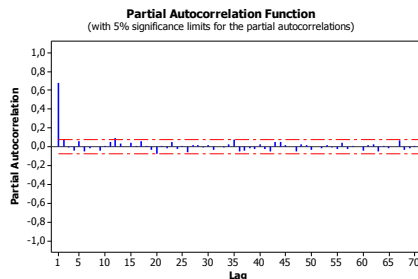
Gambar 4.9 ACF Plot Data Tree Rings

Secara visual, dilihat dari time series plot yang ada di Gambar 4.7 dan ACF plot yang ada di Gambar 4.9 menandakan data sudah stasioner terhadap mean, ditunjukkan dengan nilai observasi yang berfluktuasi di satu nilai mean pada time series plot dan lag-lag yang turun dengan cepat pada ACF plot. Namun untuk mendapatkan hasil yang akurat dilakukan pengujian *Augmented Dickey-Fuller* dengan hasil sebagai berikut.

Tabel 4.16 *Augmented Dickey-Fuller Test* Untuk Menguji Stasioneritas Terhadap Mean

Dickey-Fuller	Lag-Order	Sig.
-7,9468	8	<0,01

Signifikansi bernilai <0.01 yaitu kurang dari nilai $\alpha = 0,05$ sehingga data *tree rings* sudah stasioner terhadap *mean*. Selanjutnya membuat PACF plot untuk menentukan model seperti yang disajikan pada Gambar 4.10 berikut.



Gambar 4.10 PACF Plot Data Tree Rings

Berdasarkan PACF plot pada Gambar 4.10 dan ACF plot pada Gambar 4.9 dapat diketahui bahwa lag PACF terpotong setelah lag ke-1 dan lag ACF cenderung *dies down*, sehingga model sementara yang ditetapkan adalah ARIMA (1,0,0).

4.5.2 Estimasi dan Signifikansi Parameter

Selanjutnya parameter diestimasi dengan menggunakan algoritma iteratif yang menghitung estimasi *least squares* (Box G. J., 1994). Hasil estimasi parameter dan signifikansinya adalah sebagai berikut.

Tabel 4.17 Signifikansi Parameter

Parameter	Koefisien	T	Sig.
ϕ	0,6844	24,78	0,000

Nilai signifikansi bernilai 0,000 itu kurang dari nilai $\alpha = 0,05$ sehingga koefisien parameter signifikan untuk model ARIMA (1,0,0).

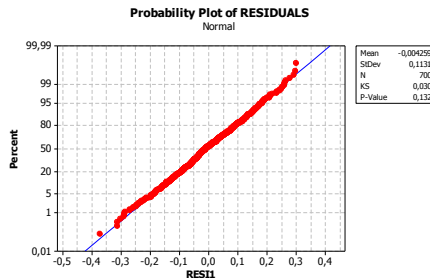
4.5.3 Diagnostic Checking

Diagnostic checking dibagi menjadi 2 pengecekan yaitu residual white noise dan residual berdistribusi normal. Berikut merupakan hasil pengecekan residual white noise.

Tabel 4.18 Pengecekan Residual *White Noise*

Lag	Chi-Square	df	Sig.
12	15,5	11	0,162
24	29,6	23	0,162
36	42,2	35	0,187
48	52,0	47	0,286

Berdasarkan Tabel 4.18 diatas residual telah memenuhi asumsi white noise karena nilai signifikansi lag-lag sudah lebih dari nilai $\alpha = 0,05$. Selanjutnya pengecekan residual berdistribusi normal ditunjukkan pada Gambar 4.11 berikut.



Gambar 4.11 *Residual Plot Data Tree Rings*

Dengan menggunakan uji *Kolmogorov Smirnov* memberikan nilai signifikansi sebesar 0,132 yaitu lebih dari nilai $\alpha = 0,05$, sehingga residual sudah memenuhi asumsi normal. Oleh karena itu, data studi kasus *tree rings* sudah tidak terdapat *outlier* di dalamnya. Namun dilakukan pengujian terhadap prosedur deteksi *outlier* yang sebelumnya dan prosedur deteksi *outlier* yang ada dalam penelitian ini untuk membandingkan efektifitas hasil deteksi *outlier*.

Dari semua tahap-tahap spesifikasi model di atas, maka data *tree rings* merupakan data dengan model ARIMA (1,0,0) dan parameter $\phi = 0,6844$. Kemudian akan dibandingkan hasil prediksi 12 periode selanjutnya apabila dilakukan ketiga cara berikut: (1) prediksi tanpa melakukan deteksi *outlier* pada data, (2) prediksi dengan melakukan deteksi *outlier* di keseluruhan data, dan (3) prediksi dengan melakukan deteksi *outlier* dan *window time*.

4.5.4 Hasil Prediksi

Berdasarkan pengujian normalitas pada sub bab sebelumnya didapatkan bahwa residual berdistribusi normal yang mengindikasikan bahwa sudah tidak terdapat *outlier* dalam data. Kemudian dilakukan perbandingan hasil prediksi dari ketiga cara. Cara deteksi *outlier* kedua, yaitu mendeteksi *outlier* pada keseluruhan data memberikan hasil deteksi *outlier* bahwa terdapat 15 *outlier* yang terdapat di dalam data. Sedangkan cara deteksi *outlier* ketiga, yaitu mendeteksi *outlier* dengan menggunakan *window time* memberikan hasil bahwa tidak terdapat *outlier* di

dalam deret data. Hasil ini sesuai dengan prediksi yang terdapat pada pengujian normalitas residual yang berdistribusi normal yang mengindikasikan tidak terdapat outlier di dalam deret data.

Ketiga cara yang dibandingkan pada data *tree rings* memberikan hasil bahwa cara pertama dan ketiga memiliki nilai RMSE yang sama karena pada hasil deteksi outlier tidak terdeteksi outlier di dalamnya sedangkan cara kedua memiliki nilai RMSE yang lebih besar dibandingkan dengan cara pertama dan ketiga, nilai RMSE masing-masing cara ditunjukkan pada Tabel 4.19 berikut

Tabel 4.19 Perbandingan RMSE Ketiga Cara Pada Data *Tree Rings*

Cara	RMSE
1	0,40891
2	0,40945
3	0,40891

Karena cara kedua memberikan nilai RMSE yang lebih besar dibandingkan cara pertama dan ketiga maka nilai prediksi deteksi *outlier* dengan menggunakan *window time* memberikan hasil yang lebih akurat dibandingkan dengan cara kedua yang tidak menggunakan deteksi *outlier window time*. Hal ini menunjukkan bahwa *outlier* yang terdeteksi pada cara kedua bukan merupakan *outlier* yang sesungguhnya sehingga terjadi misklasifikasi *outlier* apabila dilakukan cara deteksi *outlier* pada keseluruhan data yang dilakukan pada data yang panjang.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan analisis dan pembahasan pada Bab IV, maka berikut merupakan kesimpulan-kesimpulan yang dapat dipetik dari penelitian ini berkaitan dengan latar belakang dan tujuan penelitian.

Kesimpulan yang dihasilkan adalah rata-rata prosentase kesalahan deteksi *outlier* kelompok dengan lebar *window time* awal 500 tidak berbeda secara signifikan dengan rata-rata prosentase kesalahan deteksi *outlier* kelompok dengan lebar *window time* awal 1000. Sedangkan rata-rata prosentase kesalahan deteksi *outlier* kelompok dengan lebar *window time* awal 100 berbeda secara signifikan dengan rata-rata prosentase kesalahan deteksi *outlier* kelompok dengan lebar *window time* awal 500 dan 1000. Karena nilai prosentase lebar *window time* awal 100 lebih besar dibandingkan dengan nilai prosentase lebar *window time* awal 500 dan 1000, maka lebar *window time* awal 500 dan 1000 memberikan akurasi yang lebih baik. Dengan begitu, semakin kecil lebar *window* yang dipilih tidak menghasilkan keakuratan hasil deteksi *outlier* yang lebih baik.

Hasil perbandingan akurasi ketiga cara memberikan kesimpulan bahwa prediksi dengan melakukan deteksi *outlier* dan *window time* memiliki nilai RMSE yang lebih kecil dibandingkan dua cara yang lain yaitu prediksi tanpa menggunakan deteksi *outlier* dan prediksi dengan menggunakan deteksi *outlier* namun tidak menggunakan strategi *window time*. Pada percobaan dengan menggunakan data yang sebenarnya didapatkan hasil bahwa dengan menggunakan deteksi *outlier* sekaligus *window time* dan tanpa menggunakan deteksi *outlier* menghasilkan nilai RMSE yang sama dan lebih baik dibandingkan menggunakan deteksi *outlier* tanpa *window time*.

5.2 Saran

Penelitian yang telah dilakukan tentunya memiliki beberapa kekurangan, sehingga dari penelitian ini disarankan beberapa hal berikut untuk penelitian selanjutnya.

1. Sebelum menentukan model ARIMA apa yang akan dilakukan percobaan, sebaiknya melihat ketersediaan data riil terlebih dahulu. Selain itu panjang data yang akan diteliti juga harus mempertimbangkan panjang data riil yang sebelumnya sudah diperoleh.
2. Pada penelitian selanjutnya disarankan untuk meneliti lebar window antara 500 dan 1000 karena, diduga rentang lebar window tersebut menghasilkan nilai prosentase akurasi yang optimal.
3. Perlu dilakukan kombinasi parameter yang lebih beragam lagi, mengingat dalam penelitian ini terdapat satu parameter yang tidak menghasilkan kesimpulan yang sama dengan ketiga parameter yang diujicobakan.
4. Pembuatan syntax dilakukan lebih teliti dan menyeluruh.

DAFTAR PUSTAKA

- Andrews, B. H., Dean, M. D., Swain, R., & Cole, C. (2013). *Building ARIMA and ARIMAX Models for Predicting Long-Term Disability Benefit Application Rates in The Public/Private Sectors*. Portland: University of Southern Maine.
- Atok, R. M., Zaharim, A., Wahab, D. A., Mukhlisin, M., Abdullah, S., & Khatimin, N. (2015). Temporary Change Detection on ARMA(1,1) Data. *International Journal of Mathematical Models and Methods in Applied Sciences*, 9, 651-658.
- Barnett, V., & Lewis, T. (1994). *Outliers in Statistical Data* (3rd ed.). New York: John Wiley & Sons.
- Bowerman, B., O'Connell, R. T., & Koehler, A. B. (2004). *Forecasting, Time Series, and Regression: An Applied Approach*. Boston: Cengage Learning, Inc.
- Box, G. J. (1994). *Time Series Analysis Forecasting and Control* (3rd edition ed.). Englewood Cliffs: Prentice Hall.
- Box, G., & Cox, D. R. (1964). An Analysis of Transformatons. *Journal of The Royal Statistical Society, Series B (Methodological)*, 211-252.
- Box, G., Jenkins, G., & Reinsel, G. (2008). *Time Series Analysis : Forecasting and Control*. New York: John Wiley & Sons Inc.
- Brockwell, P., & Davis, R. (1996). *Introduction to Time Series and Forecasting*. New York: Springer.
- Chang, I., & Tiao, G. (1983). *Estimation of Time Series Parameters in The Presence of Outliers*. University of Chicago: Statistics Research Center.
- Chen, C., & Liu, L. M. (1993). Joint Estimation of Model Parameters and Outlier Effect in Time Series. *Journal of the American Statistical Association*, 88, 284-297.

- Cryer, J. (1986). *Time Series Analysis*. Boston: Publishing Company.
- Cryer, J. D., & Chan, K. S. (2008). *Time Series Analysis With Application in R*. New York: Springer Science.
- Durbin, J., & Koopman, S. (2001). *Time Series Analysis by State Space Methods*. Oxford University Press.
- Fox, A. J. (1972). Outliers in Time Series. *Wiley for the Royal Statistical Society*, 34, 350-363.
- Gardner, G., Harvey, A., & Philips, G. (1980). Algorithm AS154. An Algorithm for Exact Maximum Likelihood Estimation of Autoregressive-Moving Average Models by Mean of Kalman Filtering. *Applied Statistics*, 311-322.
- Ghozali, I. (2009). *Aplikasi Analisis Multivariate dengan Program SPSS*. Semarang: UNDIP.
- Hadi, A. F. (2016). *Model Hibrida Kombinasi ARIMAX-NN dan GARCH untuk Peramalan Inflow dan Outflow Uang Kartal*. Surabaya.
- Hanke, J. E., & Wichern, D. W. (2005). *Business Forecasting*. New Jersey: Prentice Hall.
- Harvey, A. (1993). *Time Series Models*, 2nd Edition. Harvester Wheatsheaf.
- Hyndman, R., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The Forecast Package for R. *Journal of Statistical Software*, 26(3).
- Jones, R. (1980). Maximum Likelihood Fitting of ARMA Models to Time Series with Missing Observationa. *Technometrics*, 389-395.
- Klinkenberg, R. (2004). Learning Drifting Concept: Example Selection vs Example Weighting. *Intelligent Data Analysis*, 281-300.
- Lenny Budiarti, T. B. (2013). *Analisis Intervensi dan Deteksi Outlier pada Data Wisatawan Domestik (Studi Kasus di Daerah Istimewa Yogyakarta)*. Yogyakarta: Jurnal Gaussian.

- Makridakis, S. S. (1992). *Metode dan Aplikasi Peramalan - Edisi ke-2 Jilid I. Alih Bahasa: Andriyanto* (Edisi ke-2 Jilid I ed.). Jakarta: Erlangga.
- Soejoeti, Z. (1987). *Analisis Runtun Waktu, Materi Pokok UT*. Jakarta: Karunika.
- Suhartono. (2007). *Teori dan Aplikasi Model Intervensi Fungsi Pulse*. Surabaya.
- Sun, J., & Li, H. (2011). Dynamic financial distress prediction using instance selection for the disposal. *Expert System with Application* 38, 2566-2576.
- Taylor, J. W., & McSharry, P. E. (2008). Short-Term Load Forecasting Methods: An Evaluation Based on European Data. *IEEE Transaction on Power System*, 22, 2213-2219.
- Tsay, R. S. (1986, Mar). Time Series Model Specification in the Presence of Outliers. *Journal of the American Statistical Association*, No. 393, 81, 132-140.
- Tsay, R. S. (1988). Outliers, Level Shifts, and Variance Changes in Time Series. *Journal of Forecasting*, 7, 1-20.
- Wei, W. (2006). *Time Series Analysis, Univariate and Multivariate Methods*. New York: Pearson Education.
- Widmer, G., & Kubat, M. (1996). Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, 69-101.

(halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1. Statistika Deskriptif Prosentase Kesalahan Deteksi *Outlier*

paramater				Mean	Std. Deviation	N
-0,8	AO	100	Belakang	0,058	0,049	100
			Depan	0,075	0,044	100
			Tengah	0,052	0,045	100
			Total	0,062	0,047	300
		500	Belakang	0,058	0,049	100
			Depan	0,073	0,044	100
			Tengah	0,051	0,045	100
			Total	0,061	0,047	300
		1000	Belakang	0,057	0,049	100
			Depan	0,074	0,045	100
			Tengah	0,050	0,044	100
			Total	0,060	0,047	300
		Total	Belakang	0,057	0,048	300
			Depan	0,074	0,044	300
			Tengah	0,051	0,045	300
			Total	0,061	0,047	900
	IO	100	Belakang	0,028	0,015	100
			Depan	0,038	0,026	100
			Tengah	0,034	0,021	100
			Total	0,033	0,022	300
		500	Belakang	0,027	0,014	100
			Depan	0,038	0,027	100
			Tengah	0,033	0,021	100

		1000	Total	0,033	0,022	300
			Belakang	0,026	0,013	100
			Depan	0,037	0,027	100
			Tengah	0,032	0,020	100
			Total	0,032	0,021	300
		Total	Belakang	0,027	0,014	300
			Depan	0,038	0,027	300
			Tengah	0,033	0,021	300
			Total	0,033	0,021	900
	TC	100	Belakang	0,021	0,000	100
			Depan	0,034	0,021	100
			Tengah	0,029	0,018	100
			Total	0,028	0,017	300
		500	Belakang	0,021	0,000	100
			Depan	0,034	0,022	100
			Tengah	0,021	0,000	100
			Total	0,025	0,014	300
		1000	Belakang	0,022	0,005	100
			Depan	0,033	0,021	100
			Tengah	0,021	0,000	100
			Total	0,025	0,014	300
		Total	Belakang	0,021	0,003	300
			Depan	0,034	0,021	300
			Tengah	0,024	0,011	300
			Total	0,026	0,015	900
	Total	100	Belakang	0,036	0,033	300
			Depan	0,049	0,037	300
			Tengah	0,038	0,032	300
			Total	0,041	0,035	900

		500	Belakang	0,035	0,033	300
			Depan	0,048	0,037	300
			Tengah	0,035	0,031	300
			Total	0,040	0,034	900
		1000	Belakang	0,035	0,033	300
			Depan	0,048	0,037	300
			Tengah	0,034	0,031	300
			Total	0,039	0,034	900
		Total	Belakang	0,035	0,033	900
			Depan	0,048	0,037	900
			Tengah	0,036	0,031	900
			Total	0,040	0,034	2700
-0,5	AO	100	Belakang	0,049	0,101	100
			Depan	0,068	0,095	100
			Tengah	0,051	0,148	100
			Total	0,056	0,117	300
		500	Belakang	0,047	0,097	100
			Depan	0,059	0,087	100
			Tengah	0,050	0,148	100
			Total	0,052	0,114	300
		1000	Belakang	0,047	0,097	100
			Depan	0,064	0,088	100
			Tengah	0,050	0,148	100
			Total	0,053	0,114	300
		Total	Belakang	0,048	0,098	300
			Depan	0,064	0,090	300
			Tengah	0,050	0,148	300
			Total	0,054	0,115	900
	IO	100	Belakang	0,045	0,035	100

			Depan	0,051	0,100	100
			Tengah	0,021	0,002	100
			Total	0,039	0,062	300
		500	Belakang	0,029	0,021	100
			Depan	0,022	0,005	100
			Tengah	0,021	0,002	100
			Total	0,024	0,013	300
		1000	Belakang	0,027	0,017	100
			Depan	0,022	0,005	100
			Tengah	0,021	0,002	100
			Total	0,023	0,011	300
		Total	Belakang	0,034	0,027	300
			Depan	0,031	0,059	300
			Tengah	0,021	0,002	300
			Total	0,029	0,038	900
	TC	100	Belakang	0,021	0,000	100
			Depan	0,048	0,095	100
			Tengah	0,045	0,035	100
			Total	0,038	0,060	300
		500	Belakang	0,021	0,000	100
			Depan	0,021	0,002	100
			Tengah	0,021	0,000	100
			Total	0,021	0,001	300
		1000	Belakang	0,022	0,004	100
			Depan	0,021	0,002	100
			Tengah	0,021	0,000	100
			Total	0,021	0,002	300
		Total	Belakang	0,021	0,002	300
			Depan	0,030	0,056	300

			Tengah	0,029	0,023	300
			Total	0,027	0,035	900
	Total	100	Belakang	0,038	0,062	300
			Depan	0,056	0,097	300
			Tengah	0,039	0,089	300
			Total	0,044	0,084	900
		500	Belakang	0,032	0,058	300
			Depan	0,034	0,053	300
			Tengah	0,031	0,086	300
			Total	0,032	0,068	900
		1000	Belakang	0,032	0,057	300
			Depan	0,036	0,054	300
			Tengah	0,031	0,086	300
			Total	0,033	0,068	900
		Total	Belakang	0,034	0,059	900
			Depan	0,042	0,072	900
			Tengah	0,033	0,087	900
			Total	0,037	0,074	2700
0,5	AO	100	Belakang	0,025	0,021	100
			Depan	0,049	0,024	100
			Tengah	0,026	0,021	100
			Total	0,033	0,025	300
		500	Belakang	0,023	0,021	100
			Depan	0,048	0,024	100
			Tengah	0,025	0,022	100
			Total	0,032	0,025	300
		1000	Belakang	0,023	0,021	100
			Depan	0,048	0,024	100
			Tengah	0,025	0,021	100

		Total	Total	0,032	0,025	300	
			Total	Belakang	0,024	0,021	300
				Depan	0,048	0,024	300
				Tengah	0,026	0,021	300
				Total	0,033	0,025	900
	IO	100	Belakang	0,027	0,012	100	
			Depan	0,035	0,017	100	
			Tengah	0,031	0,016	100	
			Total	0,031	0,015	300	
		500	Belakang	0,026	0,012	100	
			Depan	0,024	0,008	100	
			Tengah	0,030	0,016	100	
			Total	0,027	0,012	300	
		1000	Belakang	0,025	0,011	100	
			Depan	0,033	0,017	100	
			Tengah	0,030	0,016	100	
			Total	0,029	0,015	300	
		Total	Belakang	0,026	0,012	300	
			Depan	0,031	0,015	300	
			Tengah	0,030	0,016	300	
			Total	0,029	0,014	900	
	TC	100	Belakang	0,021	0,000	100	
			Depan	0,032	0,016	100	
			Tengah	0,028	0,013	100	
			Total	0,027	0,013	300	
		500	Belakang	0,021	0,000	100	
			Depan	0,031	0,016	100	
			Tengah	0,021	0,000	100	
			Total	0,024	0,011	300	

		1000	Belakang	0,022	0,005	100
			Depan	0,031	0,016	100
			Tengah	0,021	0,000	100
			Total	0,025	0,011	300
		Total	Belakang	0,021	0,003	300
			Depan	0,031	0,016	300
			Tengah	0,023	0,008	300
			Total	0,025	0,012	900
	Total	100	Belakang	0,024	0,014	300
			Depan	0,039	0,021	300
			Tengah	0,029	0,017	300
			Total	0,031	0,019	900
		500	Belakang	0,024	0,014	300
			Depan	0,035	0,020	300
			Tengah	0,026	0,016	300
			Total	0,028	0,018	900
		1000	Belakang	0,024	0,014	300
			Depan	0,037	0,021	300
			Tengah	0,025	0,016	300
			Total	0,029	0,018	900
		Total	Belakang	0,024	0,014	900
			Depan	0,037	0,021	900
			Tengah	0,026	0,016	900
			Total	0,029	0,018	2700
0,8	AO	100	Belakang	0,060	0,107	100
			Depan	0,076	0,110	100
			Tengah	0,036	0,064	100
			Total	0,057	0,097	300
		500	Belakang	0,059	0,107	100

			Depan	0,082	0,107	100
			Tengah	0,035	0,063	100
			Total	0,059	0,096	300
		1000	Belakang	0,058	0,107	100
			Depan	0,081	0,107	100
			Tengah	0,035	0,063	100
			Total	0,058	0,096	300
		Total	Belakang	0,059	0,107	300
			Depan	0,080	0,107	300
			Tengah	0,035	0,063	300
			Total	0,058	0,096	900
	IO	100	Belakang	0,035	0,051	100
			Depan	0,040	0,053	100
			Tengah	0,034	0,050	100
			Total	0,036	0,051	300
		500	Belakang	0,034	0,051	100
			Depan	0,022	0,005	100
			Tengah	0,033	0,050	100
			Total	0,030	0,041	300
		1000	Belakang	0,028	0,048	100
			Depan	0,036	0,053	100
			Tengah	0,033	0,050	100
			Total	0,032	0,050	300
		Total	Belakang	0,033	0,050	300
			Depan	0,033	0,044	300
			Tengah	0,033	0,050	300
			Total	0,033	0,048	900
	TC	100	Belakang	0,023	0,008	100
			Depan	0,034	0,050	100

			Tengah	0,041	0,065	100
			Total	0,033	0,048	300
		500	Belakang	0,021	0,000	100
			Depan	0,034	0,050	100
			Tengah	0,021	0,000	100
			Total	0,025	0,030	300
		1000	Belakang	0,021	0,000	100
			Depan	0,034	0,050	100
			Tengah	0,021	0,000	100
			Total	0,025	0,030	300
		Total	Belakang	0,022	0,005	300
			Depan	0,034	0,050	300
			Tengah	0,028	0,039	300
			Total	0,028	0,037	900
	Total	100	Belakang	0,040	0,070	300
			Depan	0,050	0,078	300
			Tengah	0,037	0,060	300
			Total	0,042	0,070	900
		500	Belakang	0,038	0,070	300
			Depan	0,046	0,073	300
			Tengah	0,030	0,047	300
			Total	0,038	0,064	900
		1000	Belakang	0,036	0,069	300
			Depan	0,050	0,078	300
			Tengah	0,030	0,047	300
			Total	0,039	0,066	900
		Total	Belakang	0,038	0,070	900
			Depan	0,049	0,076	900
			Tengah	0,032	0,052	900

			Total	0,040	0,067	2700
Total	AO	100	Belakang	0,048	0,079	400
			Depan	0,067	0,077	400
			Tengah	0,041	0,085	400
			Total	0,052	0,081	1200
		500	Belakang	0,047	0,078	400
			Depan	0,066	0,074	400
			Tengah	0,040	0,085	400
			Total	0,051	0,080	1200
		1000	Belakang	0,046	0,078	400
			Depan	0,067	0,074	400
			Tengah	0,040	0,085	400
			Total	0,051	0,080	1200
		Total	Belakang	0,047	0,078	1200
			Depan	0,067	0,075	1200
			Tengah	0,041	0,085	1200
			Total	0,051	0,080	3600
	IO	100	Belakang	0,034	0,033	400
			Depan	0,041	0,059	400
			Tengah	0,030	0,029	400
			Total	0,035	0,043	1200
		500	Belakang	0,029	0,029	400
			Depan	0,026	0,016	400
			Tengah	0,030	0,029	400
			Total	0,028	0,025	1200
		1000	Belakang	0,027	0,027	400
			Depan	0,032	0,031	400
			Tengah	0,029	0,029	400
			Total	0,029	0,029	1200

		Total	Belakang	0,030	0,030	1200
			Depan	0,033	0,040	1200
			Tengah	0,029	0,029	1200
			Total	0,031	0,033	3600
	TC	100	Belakang	0,022	0,004	400
			Depan	0,037	0,056	400
			Tengah	0,036	0,039	400
			Total	0,032	0,040	1200
		500	Belakang	0,021	0,000	400
			Depan	0,030	0,029	400
			Tengah	0,021	0,000	400
			Total	0,024	0,017	1200
		1000	Belakang	0,022	0,004	400
			Depan	0,030	0,029	400
			Tengah	0,021	0,000	400
			Total	0,024	0,017	1200
		Total	Belakang	0,021	0,003	1200
			Depan	0,032	0,040	1200
			Tengah	0,026	0,024	1200
			Total	0,027	0,027	3600
	Total	100	Belakang	0,034	0,051	1200
			Depan	0,048	0,066	1200
			Tengah	0,036	0,057	1200
			Total	0,040	0,058	3600
		500	Belakang	0,032	0,049	1200
			Depan	0,041	0,050	1200
			Tengah	0,030	0,052	1200
			Total	0,034	0,051	3600
		1000	Belakang	0,031	0,049	1200

			Depan	0,043	0,052	1200
			Tengah	0,030	0,052	1200
			Total	0,035	0,051	3600
		Total	Belakang	0,033	0,049	3600
			Depan	0,044	0,057	3600
			Tengah	0,032	0,054	3600
			Total	0,036	0,054	10800

Lampiran 2. Hasil Pengujian ANOVA Prosentase Kesalahan
Deteksi *Outlier*

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	2,623 ^a	107	0,025	9,227	0
Intercept	14,191	1	14,191	5342,121	0
paramater	0,207	3	0,069	25,995	0
outlier	1,263	2	0,631	237,658	0
panjang_initial	0,06	2	0,03	11,209	0
lokasi	0,323	2	0,161	60,765	0
paramater * outlier	0,251	6	0,042	15,74	0
paramater * panjang_initial	0,041	6	0,007	2,552	0,018
paramater * lokasi	0,029	6	0,005	1,801	0,095
outlier * panjang_initial	0,018	4	0,005	1,716	0,143
outlier * lokasi	0,196	4	0,049	18,47	0
panjang_initial * lokasi	0,01	4	0,002	0,933	0,443
paramater * outlier * panjang_initial	0,015	12	0,001	0,473	0,932
paramater * outlier * lokasi	0,116	12	0,01	3,651	0
paramater * panjang_initial * lokasi	0,025	12	0,002	0,774	0,678
outlier * panjang_initial * lokasi	0,044	8	0,006	2,083	0,034
paramater * outlier * panjang_initial * lokasi	0,025	24	0,001	0,398	0,996
Error	28,402	10692	0,003		
Total	45,215	10800			
Corrected Total	31,024	10799			

Lampiran 3. Hasil Pengujian Tukey Prosentase Kesalahan
Deteksi *Outlier*

(I) panjang_initial	(J) panjang_initial	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
100	500	,00512*	0,001215	0	0,00227	0,00797
	1000	,00483*	0,001215	0	0,00198	0,00768
500	100	-,00512*	0,001215	0	-0,00797	-0,00227
	1000	-0,00029	0,001215	0,969	-0,00314	0,00256
1000	100	-,00483*	0,001215	0	-0,00768	-0,00198
	500	0,00029	0,001215	0,969	-0,00256	0,00314

Lampiran 4. Data Simulasi Model ARIMA (1,0,0) dengan parameter $\phi = 0.8$

	rep1	rep2	rep3	rep4	rep5	rep6	rep7	rep8	...	rep100
101	-0,094	0,914	-0,256	0,339	-1,342	2,183	-0,785	-2,168	...	-0,486
102	-0,138	1,670	-0,476	-0,595	-0,898	3,164	-1,325	-0,821	...	-1,657
103	-0,339	-0,128	-0,808	-0,070	-1,088	1,122	-2,426	1,048	...	-0,914
104	0,305	0,383	-2,144	0,930	-0,638	1,614	-3,372	0,214	...	-1,171
105	0,185	1,789	-3,219	-0,603	-1,052	2,218	-3,205	1,445	...	1,024
106	0,016	0,795	-2,097	0,027	-1,738	1,696	-2,421	2,339	...	0,437
107	-0,791	-1,588	-3,459	-0,291	-3,199	0,295	-0,815	2,727	...	2,747
108	-0,360	-1,417	-1,760	1,480	-3,078	-0,952	1,248	2,404	...	2,993
109	0,771	-2,964	-0,883	1,605	-1,092	-2,766	0,991	1,096	...	3,005
110	0,148	-0,546	-1,638	1,176	-0,800	-2,790	-0,062	0,194	...	3,253
111	1,820	-1,228	-0,662	0,960	-0,183	-3,177	1,035	0,629	...	3,900
112	1,253	-1,902	-0,043	0,792	0,048	-3,909	0,312	-0,186	...	3,044
113	0,888	-0,537	1,211	0,337	0,493	-3,379	0,984	1,136	...	1,475
114	2,646	0,261	1,316	0,387	0,900	-1,622	2,367	2,017	...	2,068
115	3,388	0,741	-0,151	-0,008	-0,039	-1,792	2,862	3,318	...	0,650
116	1,742	0,320	1,125	0,950	-0,018	-2,654	2,747	0,856	...	1,559
117	1,817	0,425	1,209	1,301	-0,226	-2,942	2,436	1,887	...	-0,452
118	0,777	1,212	1,553	-0,552	3,087	-2,619	1,221	2,211	...	0,283
119	-0,262	-0,297	-0,279	-0,516	0,991	-3,334	1,136	0,417	...	-0,658
120	-0,361	0,288	-0,552	0,837	0,626	-2,321	1,271	-1,966	...	0,729
121	0,007	0,971	-0,174	0,583	2,220	-1,590	2,171	-0,396	...	-0,271
122	-0,184	0,821	-0,223	1,315	1,676	-1,248	2,052	-1,353	...	0,918
123	0,081	0,919	-0,605	1,757	-0,170	-0,630	0,646	-1,011	...	-0,565
124	1,868	1,559	0,345	1,643	-1,978	-0,043	0,870	-2,245	...	-0,803
...
5000	0,990	0,403	-0,940	-0,254	-2,997	-0,761	1,328	0,946	...	-1,698

Lampiran 5. Data Simulasi Model ARIMA (1,0,0) dengan parameter $\phi = -0.8$

	rep1	rep2	rep3	rep4	rep5	rep6	rep7	rep8	rep9	rep10
101	1,570	1,346	1,857	-2,025	-0,208	0,288	1,510	2,017	...	1,966
102	-0,734	-1,727	0,171	1,486	-0,429	0,281	-1,220	-2,033	...	-1,599
103	-0,639	1,667	-1,898	-0,794	1,663	-0,209	0,944	2,381	...	1,886
104	-0,141	-2,028	1,010	1,004	-0,879	-0,091	-0,322	-1,807	...	-1,127
105	-0,685	0,541	-1,948	0,392	0,232	0,323	0,493	2,478	...	0,782
106	1,762	0,567	-0,497	-0,900	2,070	-1,445	-0,687	-3,451	...	-0,544
107	-2,085	-0,920	-0,756	1,318	-1,634	-1,094	0,425	4,038	...	-0,818
108	2,881	-0,727	-1,727	-0,178	2,537	1,747	0,358	-2,051	...	1,191
109	-1,876	-0,144	2,751	-0,770	-1,112	-1,482	0,689	2,772	...	0,053
110	1,002	0,212	-2,460	0,820	1,685	2,090	0,757	-1,449	...	0,311
111	-1,348	0,315	3,128	0,145	-2,122	-0,371	-1,018	1,284	...	0,396
112	2,504	-1,017	-1,703	-1,939	2,174	0,369	0,667	-0,931	...	0,377
113	-3,007	2,204	1,546	2,343	-2,771	-0,032	0,229	1,242	...	-0,897
114	2,036	-2,873	-1,152	-3,125	1,575	0,422	-0,462	-0,450	...	0,997
115	-1,814	1,435	0,237	2,760	-2,112	-0,980	0,133	-0,436	...	-1,960
116	1,380	0,202	0,994	-2,112	-0,349	0,759	-0,571	0,485	...	-0,451
117	-0,298	-0,071	-2,091	1,044	1,087	-0,160	-0,001	-0,602	...	-0,057
118	1,460	-2,072	1,856	-1,478	0,740	-0,689	-1,281	3,001	...	-0,108
119	-2,196	1,539	0,651	0,918	-2,979	2,275	-0,002	-3,604	...	0,147
120	3,174	-0,031	0,444	0,444	4,689	-1,647	-0,342	2,182	...	0,988
121	-1,657	1,153	0,929	-0,100	-2,931	0,915	-1,281	-1,090	...	0,494
122	0,172	-1,756	-1,367	0,776	3,780	-1,295	0,298	1,896	...	-1,783
123	-0,206	2,164	1,296	-0,721	-1,671	1,423	0,582	-3,221	...	3,919
124	-0,111	-1,697	-0,159	-0,592	1,205	-1,648	-3,388	2,569	...	-3,219
...
5000	-2,746	0,853	2,187	2,259	-1,250	0,162	0,336	-1,699	...	1,335

Lampiran 6. Data Simulasi Model ARIMA (1,0,0) dengan parameter $\phi = 0.5$

	rep1	rep2	rep3	rep4	rep5	rep6	rep7	rep8	rep9	rep10
101	0,254	-0,538	-0,122	-0,481	2,733	1,130	1,126	1,950	...	3,880
102	-1,103	-0,885	-0,965	0,423	1,657	-0,133	-0,992	1,413	...	2,569
103	-2,301	0,379	-1,258	0,954	0,661	-1,547	0,473	-1,398	...	-0,304
104	-0,396	-0,598	-0,897	-2,485	1,267	-1,193	0,556	-2,126	...	-0,892
105	-0,492	2,363	-1,262	-3,452	0,659	0,595	-0,119	-0,134	...	-1,467
106	1,018	2,847	-0,623	-2,158	0,560	0,345	0,348	-0,640	...	0,569
107	0,517	2,010	-1,178	-1,683	0,196	1,145	1,787	0,035	...	-0,107
108	1,083	0,920	-0,622	-1,342	-1,401	0,929	-0,337	0,567	...	0,931
109	-0,162	1,684	-0,095	-1,193	-1,976	1,784	-0,074	-0,928	...	0,142
110	0,531	0,577	0,205	-1,148	-0,438	2,096	0,058	-0,243	...	-0,490
111	2,163	-1,247	-1,018	-0,936	0,668	0,675	-1,164	-1,018	...	0,545
112	0,576	-0,957	-1,393	-1,540	-1,961	-0,096	-1,783	-0,687	...	0,590
113	0,750	-0,955	-1,375	0,975	-0,722	0,206	-0,132	-0,831	...	0,366
114	-0,989	1,807	-0,364	-0,071	-0,409	0,897	-0,352	-0,005	...	1,940
115	1,100	0,546	-2,443	0,037	0,666	0,100	-1,594	1,522	...	1,183
116	2,097	0,340	-0,813	0,904	-0,354	-0,292	-3,131	0,443	...	0,162
117	-0,305	1,132	0,011	1,235	-2,843	0,658	-1,451	-0,603	...	1,932
118	-0,014	1,323	-0,222	0,604	-1,099	1,033	-1,597	-0,991	...	-0,952
119	1,441	0,333	0,175	-0,216	-1,236	-0,277	-2,904	-1,902	...	-0,717
120	0,951	0,957	1,261	-0,096	0,570	-2,972	-3,348	-1,376	...	0,789
121	0,769	0,540	-0,438	-1,532	-1,895	-1,148	0,076	1,185	...	1,416
122	-1,213	0,330	-0,858	-1,928	-0,711	-0,394	-0,696	1,546	...	0,702
123	0,260	1,465	-1,603	-2,238	0,694	0,151	0,259	1,035	...	1,929
124	2,148	0,960	-2,307	-0,694	1,505	-0,020	0,302	-0,287	...	-0,558
...
5000	-0,601	-0,570	1,601	0,019	-0,745	0,725	0,430	0,855	...	0,422

Lampiran 7. Data Simulasi Model ARIMA (1,0,0) dengan parameter $\phi = -0.5$

	rep1	rep2	rep3	rep4	rep5	rep6	rep7	rep8	rep9	rep10
101	0,731	-1,523	0,925	-0,108	-0,499	-1,229	1,470	-0,403	...	-0,861
102	-0,531	0,937	0,611	-1,341	-0,664	2,018	-1,622	1,136	...	0,119
103	-1,430	0,386	0,350	1,187	-0,263	-1,127	-0,233	-1,772	...	-0,820
104	0,794	-1,236	-1,214	-1,389	0,969	-0,544	2,532	1,377	...	-1,867
105	-0,529	0,857	0,493	0,446	-0,441	-0,980	-0,955	-2,228	...	0,419
106	0,670	-0,651	0,812	-0,737	-0,036	0,728	1,088	0,995	...	-1,306
107	1,465	1,589	-1,578	2,633	0,157	-0,621	-1,284	-1,626	...	0,417
108	-0,712	0,422	0,718	-0,997	0,993	-1,264	-0,812	0,409	...	-0,153
109	1,019	-0,597	-0,676	-1,469	-1,695	0,213	-0,317	0,582	...	-1,239
110	-0,690	0,664	0,368	1,341	-0,096	0,272	-0,476	-0,541	...	1,289
111	1,139	0,415	-0,766	-1,172	-1,082	1,153	0,058	0,007	...	-0,307
112	-1,843	-0,952	-0,570	0,323	-0,450	-0,834	-0,256	-1,081	...	-0,067
113	0,789	1,022	0,761	0,745	0,627	-0,840	-0,272	4,324	...	-0,833
114	-0,655	-0,173	-0,685	-2,526	1,492	0,139	1,491	-2,451	...	1,132
115	2,585	2,072	0,468	3,831	1,047	1,058	-0,334	-0,947	...	-0,470
116	-2,123	-0,892	-0,935	-1,768	-0,543	0,063	1,367	-0,149	...	-0,014
117	1,012	-0,460	0,041	1,200	-0,709	0,400	-0,818	0,465	...	-0,548
118	1,752	0,434	-1,565	0,361	-0,446	2,045	0,053	0,083	...	-0,940
119	-0,015	-1,396	1,987	-2,071	0,754	-1,243	1,273	-0,755	...	-1,228
120	1,035	1,582	1,018	-0,984	-1,231	0,524	0,943	1,128	...	0,521
121	0,560	-0,264	-0,743	0,824	0,370	-0,656	-0,728	-1,827	...	-0,457
122	-0,535	1,629	1,258	-1,815	0,758	0,139	0,139	1,450	...	-3,230
123	0,246	-0,995	-0,327	2,546	-1,355	0,090	-0,744	-1,919	...	0,453
124	1,179	-0,716	0,723	-2,512	1,130	0,100	0,363	3,044	...	0,510
...
5000	1,275	-1,583	-0,824	-1,617	-0,462	1,896	0,679	-1,294	...	-0,972

Lampiran 8. Prosentase Kesalahan Deteksi *Outlier* Kombinasi Parameter, Jenis *Outlier*, Lebar Awal *Window Time* dan Lokasi *Outlier*

No	Stacked	Parameter	Jenis Outlier	Panjang initial	Lokasi Outlier
1	0,021	0,8	AO	100	Belakang
2	0,042	0,8	AO	100	Belakang
...
100	0,021	0,8	AO	100	Belakang
101	0,021	-0,8	AO	100	Belakang
102	0,083	-0,8	AO	100	Belakang
...
200	0,042	-0,8	AO	100	Belakang
201	0,000	0,5	AO	100	Belakang
202	0,083	0,5	AO	100	Belakang
...
300	0,042	0,5	AO	100	Belakang
301	0,000	-0,5	AO	100	Belakang
302	0,042	-0,5	AO	100	Belakang
...
400	0,021	-0,5	AO	100	Belakang
401	0,021	0,8	AO	500	Belakang
402	0,042	0,8	AO	500	Belakang
...
500	0,021	0,8	AO	500	Belakang
501	0,021	-0,8	AO	500	Belakang
502	0,083	-0,8	AO	500	Belakang
...
600	0,042	-0,8	AO	500	Belakang
601	0,000	0,5	AO	500	Belakang
602	0,083	0,5	AO	500	Belakang
...
10800	0,021	-0,5	TC	1000	Belakang

Lampiran 9. Parameter Data Simulasi ARIMA (1,0,0)

Rep	Parameter	Rep	Parameter	Rep	Parameter	Rep	Parameter
1	0,804	26	0,790	51	0,797	76	0,785
2	0,799	27	0,807	52	0,789	77	0,799
3	0,805	28	0,803	53	0,813	78	0,799
4	0,796	29	0,807	54	0,777	79	0,800
5	0,793	30	0,801	55	0,808	80	0,797
6	0,798	31	0,796	56	0,808	81	0,794
7	0,805	32	0,785	57	0,789	82	0,786
8	0,816	33	0,805	58	0,802	83	0,797
9	0,805	34	0,813	59	0,784	84	0,786
10	0,810	35	0,790	60	0,800	85	0,791
11	0,795	36	0,789	61	0,792	86	0,804
12	0,804	37	0,808	62	0,803	87	0,798
13	0,800	38	0,794	63	0,803	88	0,800
14	0,806	39	0,796	64	0,804	89	0,800
15	0,797	40	0,799	65	0,813	90	0,812
16	0,814	41	0,802	66	0,810	91	0,792
17	0,778	42	0,793	67	0,808	92	0,788
18	0,806	43	0,808	68	0,801	93	0,796
19	0,800	44	0,794	69	0,799	94	0,789
20	0,805	45	0,798	70	0,804	95	0,797
21	0,809	46	0,809	71	0,789	96	0,800
22	0,806	47	0,812	72	0,797	97	0,783
23	0,813	48	0,795	73	0,789	98	0,799
24	0,805	49	0,815	74	0,802	99	0,788
25	0,807	50	0,806	75	0,817	100	0,802

Lampiran 10. Data *Tree Rings*

<i>Year</i>	<i>Tree Rings</i>	<i>Year</i>	<i>Tree Rings</i>	<i>Year</i>	<i>Tree Rings</i>	<i>Year</i>	<i>Tree Rings</i>
1242	1,343	1422	1,099	1602	0,957	1782	1,122
1243	1,044	1423	1,214	1603	1,206	1783	0,965
1244	1,174	1424	1,035	1604	1,074	1784	1,172
1245	1,068	1425	0,999	1605	0,953	1785	1,077
1246	0,913	1426	0,919	1606	0,922	1786	1,119
1247	0,688	1427	0,821	1607	0,791	1787	1,165
1248	0,648	1428	0,956	1608	0,802	1788	1,067
1249	0,838	1429	1,011	1609	0,982	1789	0,891
1250	1,022	1430	0,971	1610	1,109	1790	0,961
1251	0,964	1431	1,124	1611	1,031	1791	0,89
1252	1,004	1432	0,87	1612	1,048	1792	0,917
1253	0,995	1433	1,086	1613	1,086	1793	0,928
1254	1,124	1434	0,963	1614	1,164	1794	1,082
1255	1,228	1435	0,9	1615	1,207	1795	1,29
1256	1,641	1436	1,063	1616	1,125	1796	1,234
1257	1,554	1437	0,841	1617	1,216	1797	1,302
1258	1,11	1438	0,987	1618	1,086	1798	1,137
1259	1,027	1439	0,886	1619	1,067	1799	1,118
1260	0,853	1440	0,994	1620	0,955	1800	1,229
1261	1,173	1441	0,936	1621	0,949	1801	1,067
1262	0,205	1442	1,036	1622	0,918	1802	1,021
1263	0,205	1443	0,937	1623	0,95	1803	0,869
1264	1,416	1444	0,9	1624	0,826	1804	0,97
1265	1,567	1445	0,942	1625	0,887	1805	1,002
...
1421	1,054	1601	1,106	1781	1,091	1975	0,869

Lampiran 11. Surat Pernyataan Data Tugas Akhir**SURAT PERNYATAAN**

Saya yang bertanda tangan di bawah ini, mahasiswa jurusan Statistika FMIPA ITS:

Nama : Rya Sofi Aulia

NRP : 1313100064

menyatakan bahwa data yang saya gunakan dalam Tugas Akhir/Thesis ini merupakan data sekunder yang diambil dari penelitian/buku/Tugas Akhir/Thesis/publikasi lainnya yaitu:

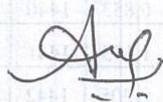
Sumber : *Time Series Data Library*

Keterangan : Tree: PIEDRA DEL AGUILA CHILE, CLIMATIC, ARAR, 1536, Location: MALLECO, ARAR, 1300M -3750-07302, Author: R.L.HOLMES, 12 26 76, Years: 1242-1975

Surat pernyataan ini dibuat dengan sebenar-benarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui,
Pembimbing Tugas Akhir

Surabaya, Januari 2017

R. Mohamad Atok, M.Si, Ph.D
NIP. 19710915 199702 1 001

Rya Sofi Aulia
NRP. 1313100064

BIODATA PENULIS



Penulis memiliki nama lengkap Rya Sofi Aulia atau biasa dipanggil dengan nama Sofi. Penulis lahir di Kabupaten Tuban pada tanggal 13 Januari 1995, namun sudah tinggal dan menempuh pendidikan wajib di Kabupaten Sidoarjo sejak berusia 3 tahun. Penulis merupakan putra tunggal dari pasangan Bapak Ansori dan Ibu Ida Zulaicha. Penulis menempuh pendidikan SD di SDN

Kalitengah 2, sedangkan pendidikan SMP di SMP Negeri 1 Sidoarjo. Kemudian penulis menempuh pendidikan SMA di SMA Negeri 1 Sidoarjo. Hingga akhirnya pada tahun 2013 penulis melanjutkan pendidikan di jenjang perguruan tinggi di jurusan Statistika ITS melalui jalur SNMPTN. Selama 3.5 tahun berkuliah di jurusan Statistika ITS, penulis juga aktif di beberapa organisasi, antara lain adalah HIMASTA-ITS 14/15, KOPMA dr. Angka ITS 2014-2016, CICA Corp dan Koperasi Pemuda Indonesia (KOPINDO). Selama aktif di organisasi tersebut penulis pernah menjabat sebagai Direktur Bidang Bisnis KOPMA dr. Angka ITS pada masa kepengurusan 2015. Segala kritik dan saran serta diskusi lebih lanjut mengenai Tugas Akhir ini dapat dikirimkan melalui surat elektronik (*e-mail*) ke ryasofia@gmail.com atau nomor telepon 085730234904.

(halaman ini sengaja dikosongkan)